Games with Delays – A Frankenstein Approach

Dietmar Berwanger and Marie van den Bogaard

CNRS & Université Paris-Saclay, ENS Cachan, LSV, France
`dwb@lsv.fr`, `mvdb@lsv.fr`

Abstract

We investigate infinite games on finite graphs where the information flow is perturbed by non-deterministic signalling delays. It is known that such perturbations make synthesis problems virtually unsolvable, in the general case. On the classical model where signals are attached to states, tractable cases are rare and difficult to identify.

In this paper, we propose a model where signals are detached from control states, and we identify a subclass on which equilibrium outcomes can be preserved, even if signals are delivered with a delay that is finitely bounded. To offset the perturbation, our solution procedure combines responses from a collection of virtual plays following an equilibrium strategy in the instant-signalling game to synthesise, in a Dr. Frankenstein manner, an equivalent equilibrium strategy for the delayed-signalling game.

1998 ACM Subject Classification C 2.4 Distributed Systems, F.1.2 Modes of Computation

Keywords and phrases infinite games on graphs, imperfect information, delayed monitoring, distributed synthesis

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2015.307

1 Introduction

Appropriate behaviour of an interactive system component often depends on events generated by other components. The ideal situation, in which perfect information is available across components, occurs rarely in practice – typically a component only receives signals more or less correlated with the actual events. Apart from imperfect signals generated by the system components, there are multiple other sources of uncertainty due to actions of the system environment or to unreliable behaviour of the infrastructure connecting the components: For instance, communication channels may delay or lose signals, or deliver them in a different order than they were emitted. Coordinating components with such imperfect information to guarantee optimal system runs is a significant, but computationally challenging, problem, in particular when the interaction is of infinite duration. It appears worthwhile to study the different sources of uncertainty in separation rather than as a global phenomenon, to understand their computational impact on the synthesis of multi-component systems.

In this paper, we consider interactive systems modelled by concurrent games among multiple players with imperfect information over finite state-transition systems, or labelled graphs. Each state is associated to a stage game in which the players choose simultaneously and independently a joint action, which triggers a transition to a successor state and generates a local payoff and possibly further private signals to each player. Plays correspond to infinite paths through the graph and yield to each player a global payoff according to a given aggregation function, such as mean payoff, limit superior payoff, or parity. As solutions to such games, we are interested in synthesising Nash equilibria in pure strategies, i.e., profiles of deterministic strategies that are self-enforcing when prescribed to all players by a central coordinator.

35th IARCS Annual Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2015).
Editors: Prahladh Harsha and G. Ramalingam; pp. 307–319
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The basic setting is standard for the automated verification and synthesis of reactive modules that maintain ongoing interaction with their environment seeking to satisfy a common global specification. Generally, imperfect information about the play is modelled as uncertainty about the current state in the underlying transition system, whereas uncertainty about the actions of other players is not represented explicitly. This is because the main question concerns *distributed* winning strategies, i.e., Nash equilibria in the special case where the players have a common utility function and should each receive maximal payoff. If every player wins when all follow the prescribed strategy, unilateral deviations cannot be profitable and any reaction to them would be ineffective, hence there is no need to monitor actions of other players. Accordingly, distributed winning strategies can be defined on (potential) histories of visited states, independently of the history of played actions. Nevertheless, these games are computationally intractable in general, already with respect to the question of whether distributed winning strategies exist [12, 11, 1].

However, if no equilibria exist that yield maximal payoffs to all players in a game, and we consider arbitrary Nash equilibria rather than distributed winning strategies, it becomes crucial for a player to monitor the actions of other players. To illustrate, one elementary scheme for constructing equilibria in games of infinite duration relies on *grim-trigger* strategies: cooperate on the prescribed equilibrium path until one player deviates, and at that event, enter a coalition with the remaining players and switch to a joint punishment strategy against the deviator. Most procedures for constructing Nash equilibria in games for verification and synthesis are based on this scheme, which relies essentially on the ability of players to detect *jointly* the deviation [15, 17, 5, 4].

The grim-trigger scheme works well under perfect, instant monitoring, where all players have common knowledge about the most recent action performed by any other player. In contrast, the situation becomes more complicated when players receive only imperfect signals about the actions of other players, and worse, if the signals are not delivered instantly, but with uncertain delays that may be different for each player. Imagine a scenario with three players, where Player 1 deviates from the equilibrium path and this is signalled to Player 2 immediately, but only with a delay to Player 3. If Player 2 triggers a punishment strategy against Player 1 as soon as she detects the deviation, Player 3 may monitor the action of Player 2 as a deviation from the equilibrium and trigger, in his turn, a punishment strategy against her, overthrowing the equilibrium outcome to the profit of Player 1.

Our contribution

We study the effect of imperfect, delayed monitoring on equilibria in concurrent games. Towards this, we first introduce a refined game model in which observations about actions are separated from observations about states, and we incorporate a representation for nondeterministic delays for observing action signals. To avoid the general undecidability results from the basic setting, we restrict to the case where the players have perfect information about the current state.

Our main result is that, under the assumption that the delays are uniformly bounded, every equilibrium payoff in the variant of a game where signals are delivered instantly is preserved as an equilibrium payoff in the variant where they are delayed. To prove this, we construct strategies for the delayed-monitoring game by combining responses for the instant-monitoring variant in such a way that any play with delayed signals corresponds to a shuffle of several plays with instant signals, which we call threads. Intuitively, delayed-monitoring strategies are constructed, in a Frankenstein manner, from a collection of instant-monitoring equilibrium strategies. Under an additional assumption that the payoff structure is insensitive to shuffling plays, this procedure allows to transfer equilibrium payoffs from the instant to the delayed-monitoring game.

Firstly, the transfer result can be regarded as an equilibrium existence theorem for games with delayed monitoring based on classes of games with instant monitoring that admit equilibria in pure strategies. Defining existence conditions is a fundamental prerequisite to using Nash equilibrium as a solution concept. If an application model leads to games that may not admit equilibria, this is a strong reason to look for another solution concept. As mixed strategies are conceptually challenging in the context of infinite games, guarantees for pure equilibrium existence are particularly desirable.

Secondly, our result establishes an outcome equivalence between games with instant and delayed monitoring, within the given restrictions: As the preservation of equilibrium values from delayed-monitoring games to the instant-monitoring variant holds trivially (the players may just buffer the received signals until an admissible delay period passed, and then respond), we obtain that the set of pure equilibrium payoffs is the same, whether signals are delayed or not — although, of course, the underlying equilibrium strategies differ between the two variants. In terms of possible equilibrium payoffs, these games are hence robust under changing signalling delivery guarantees, as long as the maximal delays are commonly known. In particular, payoff-related results obtained for the instant-signalling variant apply directly to the delayed variant.

Thirdly, the transfer procedure has some algorithmic content. When we set out with finite-state equilibrium strategies for the instant-monitoring game, the procedure will also yield a profile of finite-state strategies for the delayed-monitoring game. Hence, the construction is effective, and can be readily applied to cases where synthesis procedures for finite-state equilibria in games with instant monitoring exist.

Related literature

One motivation for studying infinite games with delays comes from the work of Shmaya [14] considering sequential games on finitely branching trees (or equivalently, on finite graphs) where the actions of players are monitored perfectly, but with arbitrary finite delays. In the setting of two-player zero-sum games with Borel winning conditions, Shmaya shows that these delayed-monitoring games are determined in mixed strategies. Apart of revealing that infinite games on finite graphs are robust under monitoring delays, the paper is enlightening for its proof technique which relies on a reduction of the delayed-monitoring game to a game with a different structure that features instant monitoring but, in exchange, involves stochastic moves.

Our analysis is inspired directly from recent work of Fudenberg, Ishii, and Kominers [8] on infinitely repeated games with bounded-delay monitoring whith stochastically distributed observation lags. The authors prove a transfer result that is much stronger than ours, which also covers the relevant case of discounted payoffs (modulo a controlled adjustment of the discount factor). The key idea for constructing strategies in the delayed-response game is to modify strategies from the instant-response game by letting them respond with a delay equal to the maximal monitoring delay so that all players received their signals. This amounts to combining different threads of the instant-monitoring game, one for every time unit in the delay period. Thus, the proof again involves a reduction between games of different structure, with the difference that here one game is reduced to several instances of another one.

Infinitely repeated games correspond to the particular case of concurrent games with only one state. This allows applying classical methods from strategic games which are no longer accessible in games with several states [13]. Additionally, the state-transition structure of our setting induces a combinatorial effort to adapt the delayed-response strategies from [8]: As the play may reach a different state until the monitoring delay expires, the instant-monitoring threads must be scheduled more carefully to make sure that they combine to a valid play of the delayed-monitoring variant. In particular, the time for returning to a particular game state may be unbounded, which makes it hard to deliver guarantees under discounted payoff functions. As a weaker notion of patience, suited for games with state transitions, we consider payoff aggregation functions that are *shift-invariant and submixing*, as introduced by Gimbert and Kelmendi in their work on memoryless strategies in stochastic games [9].

Our model generalises concurrent games of infinite duration over finite graphs. Equilibria in such models have been investigated for the perfect-information case, and it was shown that it is decidable with relatively low complexity whether equilibria exist, and if this is the case, finite-state equilibrium profiles can be synthesised for several cases of interest in automated verification. Ummels [15] considers turned-based games with parity conditions and shows that deciding whether there exists a pure Nash equilibrium payoff in a given range is an NP-complete problem. For the case of concurrent games with mean-payoff conditions, Ummels and Wojtczak [16], show that the problem for pure strategies is still NP-complete, whereas it becomes undecidable for mixed strategies. For the case of concurrent games with Büchi conditions, that is, parity conditions with priorities 1 and 2, Bouyer et al. [3] show that the complexity of the problem drops to PTime. These results are in the setting of perfect information about the actual game state and perfect monitoring. However, as pointed out in the conclusion of [3], the generic complexity increases when actions are not monitored by any player.

The basic method for constructing equilibria in the settings of perfect monitoring relies on grim-trigger strategies that react to deviations from the equilibrium path by turning to a zero-sum coalition strategy opposing the deviating player. Such an approach can hardly work under imperfect monitoring where deviating actions cannot be observed directly. Alternative approaches for constructing equilibria without relying on perfect monitoring comprise, on the one hand distributed winning strategies for games that allow all players of a coalition to attain the most efficient outcome [10, 7, 2], and at the other extreme, Doomsday equilibria, proposed by Chatterjee et al. in [6], for games where any deviation leads to the most inefficient outcome, for all players.

2 Games with delayed signals

There are n players $1, \dots, n$ and a distinguished agent called Nature. We refer to a list $x = (x^i)_{1 \le i \le n}$ that associates one element x^i to every player i as a *profile*. For any such profile, we write x^{-i} to denote the list $(x^j)_{1 \le j \le n, j \ne i}$ where the element of Player i is omitted. Given an element x^i and a list x^{-i}, we denote by (x^i, x^{-i}) the full profile $(x^i)_{1 \le i \le n}$. For clarity, we always use superscripts to specify to which player an element belongs. If not quantified explicitly, we refer to Player i to mean any arbitrary player.

2.1 General model

For every player i, we fix a set A^i of *actions*, and a set Y^i of *signals*; these sets are finite. The *action space* A consists of all action profiles, and the *signal space* Y of all signal profiles.

2.1.1 Transition structure

The transition structure of a game is described by a *game graph* $G = (V, E)$ over a finite set V of *states* with an edge relation $E \subseteq V \times A \times Y \times V$ that represents *transitions* labelled by action and signal profiles. We assume that for each state v and every action profile a, there exists at least one transition $(v, a, y, v') \in E$.

The game is played in stages over infinitely many periods starting from a designated initial state $v_0 \in V$ known to all players. In each period $t \geq 1$, starting in a state v_{t-1}, every player i chooses an action a^i_t, and Nature chooses a transition $(v_{t-1}, a_t, y_t, v_t) \in E$, which determines a profile y_t of emitted signals and a successor state v_t. Then, each player i observes a set of signals depending on the monitoring structure of the game, and the play proceeds to period $t+1$ with v_t as the new state.

Accordingly, a *play* is an infinite sequence $v_0,\ a_1, y_1, v_1,\ a_2, y_2, v_2\ \cdots \in V(AYV)^\omega$ such that $(v_{t-1}, a_t, y_t, v_t) \in E$, for all $t \geq 1$. A *history* is a finite prefix $v_0,\ a_1, y_1, v_1,\ \ldots,\ a_t, y_t, v_t \in V(AYV)^*$ of a play. We refer to the number of stages played up to period t as the *length* of the history.

2.1.2 Monitoring structure

We assume that each player i always knows the current state v and the action a^i she is playing. However, she is not informed about the actions or signals of the other players. Furthermore, she may observe the signal y^i_t emitted in a period t only in some later period or, possibly, never at all.

The signals observed by Player i are described by an *observation* function

$$\gamma^i : V(AYV)^+ \to 2^{Y^i},$$

which assigns to every nontrivial history $\pi = v_0,\ a_1, y_1, v_1,\ \ldots,\ a_t, y_t, v_t$ with $t \geq 1$, a set of signals that were actually emitted along π for the player:

$$\gamma^i(\pi) \subseteq \{y^i_r \in Y^i \mid 1 \leq r \leq t\}.$$

For an actual history $\pi \in V(AYV)^*$, the *observed history* of Player i is the sequence

$$\beta^i(\pi) := v_0,\ a^i_1, z^i_1, v_1,\ \ldots,\ a^i_t, z^i_t, v_t$$

with $z^i_r = \gamma^i(v_0,\ a_1, y_1, v_1,\ \ldots,\ a_r, y_r, v_r)$, for all $1 \leq r \leq t$. Analogously, we define the *observed play* of Player i.

A *strategy* for player i is a mapping $s^i : V(A^i 2^{Y^i} V)^* \to A^i$ that associates to every observation history $\pi \in V(A^i 2^{Y^i} V)^*$ an action $s^i(\pi)$. The *strategy space* S is the set of all strategy profiles. We say that a history or a play π *follows* a strategy s^i, if $a^i_{t+1} = s^i(\beta^i(\pi_t))$, for all histories π_t of length $t \geq 0$ in π. Likewise, a history or play follows a profile $s \in S$, if it follows the strategy s^i of each player i. The *outcome* $\mathrm{out}(s)$ of a strategy profile s is the set of all plays that follow it. Note that the outcome of a strategy profile generally consist of multiple plays, due to the different choices of Nature.

Strategies may be partial functions. However, we require that for any history π that follows a strategy s^i, the observed history $\beta^i(\pi)$ is also included in the domain of s^i.

With the above definition of a strategy, we implicitly assume that players have perfect recall, that is, they may record all the information acquired along a play. Nevertheless, in certain cases, we can restrict our attention to strategy functions computable by automata with finite memory. In this case, we speak of *finite-state strategies*.

2.1.3 Payoff structure

Every transition taken in a play generates an integer payoff to each player i, described by a *payoff* function $p^i : E \to \mathbb{Z}$. These stage payoffs are combined by a *payoff aggregation* function $u : \mathbb{Z}^\omega \to \mathbb{R}$ to determine the *utility* received by Player i in a play π as $u^i(\pi) := u(\, p^i(v_0, a_1, y_1, v_1), p^i(v_1, a_2, y_2, v_2), \dots)$. Thus, the profile of *utility*, or global payoff, functions $u^i : V(AYV)^\omega \to \mathbb{R}$ is represented by a profile of payoff functions p^i and an aggregation function u, which is common to all players.

We generally consider utilities that depend only on the observed play, that is, $u^i(\pi) = u^i(\pi')$, for any plays π, π' that are indistinguishable to Player i, that is, $\beta^i(\pi) = \beta^i(\pi')$. To extend payoff functions from plays to strategy profiles, we set

$$u^i(s) := \inf\{u^i(\pi) \mid \pi \in \text{out}(s)\}, \text{ for each strategy profile } s \in S.$$

Overall, a game $\mathcal{G} = (G, \gamma, u)$ is described by a game graph with a profile of observation functions and one of payoff functions. We are interested in *Nash equilibria*, that is, strategy profiles $s \in S$ such that $u^i(s) \geq u^i(r^i, s^{-i})$, for every player i and every strategy $r^i \in S^i$. The payoff $w = u^i(s)$ generated by an equilibrium $s \in S$ is called an equilibrium payoff. An equilibrium payoff w is *ergodic* if it does not depend on the initial state of the game, that is, there exists a strategy profile s with $u(s) = w$, for every choice of an initial state.

2.2 Instant and bounded-delay monitoring

We focus on two particular monitoring structures, one where the players observe their component of the signal profile instantly, and one where each player i observes his private signal emitted in period t in some period $t + d^i_t$, with a bounded delay $d^i_t \in \mathbb{N}$ chosen by Nature.

Formally, a game with *instant monitoring* is one where the observation functions γ^i return, for every history $\pi = v_0,\ a_1, y_1, v_1,\ \dots,\ a_t, y_t, v_t$ of length $t \geq 1$, the private signal emitted for Player i in the current stage, that is, $\gamma^i(\pi) = \{y^i_t\}$, for all $t \geq 1$. As the value is always a singleton, we may leave out the enclosing set brackets and write $\gamma^i(\pi) = y^i_t$.

To model bounded delays, we consider signals with an additional component that represents a timestamp. Concretely, we fix a set B^i of *basic signals* and a finite set $D^i \subseteq \mathbb{N}$ of *possible delays*, for each player i, and consider the product $Y^i := B^i \times D^i$ as a new set of signals. Then, a game with *delayed monitoring* is a game over the signal space Y with observation functions γ^i that return, for every history $\pi = v_0,\ a_1, (b_1, d_1), v_1,\ \dots,\ a_t, (b_t, d_t), v_t$ of length $t \geq 1$, the value

$$\gamma^i(\pi) = \{(b^i_r, d^i_r) \in B^i \times D^i \mid r \geq 1,\ r + d^i_r = t\}.$$

In our model, the role of Nature is limited to choosing the delays for observing the emitted signals. Concretely, we postulate that the basic signals and the stage payoffs associated to transitions are determined by the current state and the action profile chosen by the players, that is, for every global state v and action profile a, there exists a unique profile b of basic signals and a unique state v' such that $(v, a, (b, d), v') \in E$, for some $d \in D$; moreover, for any other delay profile $d' \in D$, we require $(v, a, (b, d'), v') \in E$, and also that $p^i(v, a, (b, d), v') = p^i(v, a, (b, d'), v')$. Here again, D denotes the *delay space* composed of the sets D^i. Notice that under this assumption, the plays in the outcome of a strategy profile s differ only by the value of the delays. In particular, all plays in $\text{out}(s)$ yield the same payoff.

To investigate the effect of observation delays, we will relate the delayed and instant-monitoring variants of a game. Given a game $\mathcal{G}$ with delayed monitoring, the corresponding instant-monitoring game $\mathcal{G}'$ is obtained by projecting every signal $y^i = (b^i, d^i)$ onto its first component b^i and then taking the transition and payoff structure induced by this projection. As we assume that transitions and payoffs are independent of delays, the operation is well defined.

Conversely, given a game $\mathcal{G}$ with instant monitoring and a delay space D, the corresponding game $\mathcal{G}'$ with delayed monitoring is obtained by extending the set B^i of basic signals in $\mathcal{G}$ to $B^i \times D^i$, for each player i, and by lifting the transition and payoff structure accordingly. Thus, the game $\mathcal{G}'$ has the same states as $\mathcal{G}$ with transitions $E' := \{(v, a, (b, d), w) \mid (v, a, b, w) \in E, d \in D\}$, whereas the payoff functions are given by ${p'}^i(v, a, (b, d), w) := p^i(v, a, b, w)$, for all $d \in D$.

As the monitoring structure of games with instant or delayed monitoring is fixed, it is sufficient to describe the game graph together with the profile of payoff functions, and to indicate the payoff aggregation function. It will be convenient to include the payoff associated to a transition as an additional edge label and thus represent the game simply as a pair $\mathcal{G} = (G, u)$ consisting of a finite labelled game graph and an aggregation function $u : \mathbb{Z}^\omega \to \mathbb{R}$.

2.3 Shift-invariant, submixing utilities

Our result applies to a class of games where the payoff-aggregation functions are invariant under removal of prefix histories and shuffling of plays. Gimbert and Kelmendi [9] identify these properties as a guarantee for the existence of simple strategies in stochastic zero-sum games.

A function $f : \mathbb{Z}^\omega \to \mathbb{R}$ is *shift-invariant*, if its value does not change when adding an arbitrary finite prefix to the argument, that is, for every sequence $\alpha \in \mathbb{Z}^\omega$ and each element $a \in \mathbb{Z}$, we have $f(a\alpha) = f(\alpha)$.

An infinite sequence $\alpha \in \mathbb{Z}^\omega$ is a *shuffle* of two sequences $\varphi, \eta \in \mathbb{Z}^\omega$, if $\mathbb{N}$ can be partitioned into two infinite sets $I = \{i_0, i_1, \dots\}$ and $J = \{j_0, j_1, \dots\}$ such that $\alpha_{i_k} = \varphi_k$ and $\alpha_{j_k} = \eta_k$, for all $k \in \mathbb{N}$. A function $f : \mathbb{Z}^\omega \to \mathbb{R}$ is called *submixing* if, for every shuffle α of two sequences $\varphi, \eta \in \mathbb{Z}^\omega$, we have

$$\min\{f(\varphi), f(\eta)\} \leq f(\alpha) \leq \max\{f(\varphi), f(\eta)\}.$$

In other words, the image of a shuffle product always lies between the images of its factors.

The proof of our theorem relies on payoff aggregation functions $u : \mathbb{Z}^\omega \to \mathbb{R}$ that are shift-invariant and submixing. Many relevant game models used in economics, game theory, and computer science satisfy this restriction. Prominent examples are mean payoff or limsup payoff, which aggregate sequences of stage payoffs $p_1, p_2, \dots \in \mathbb{Z}^\omega$ by setting:

$$\text{mean-payoff}(p_1, p_2, \dots) := \limsup_{t \geq 1} \frac{1}{t} \sum_{r=1}^{t} p_r, \quad \text{and}$$

$$\text{limsup}(p_1, p_2, \dots) := \limsup_{t \geq 1} p_t.$$

Finally, parity conditions which map non-negative integer payoffs $p_1, p_2, \dots$ called priorities to $\text{parity}(p_1, p_2, \dots) = 1$ if the least priority that occurs infinitely often is even, and 0 otherwise, also satisfy the conditions.

2.4 The transfer theorem

We are now ready to formulate our result stating that, under certain restrictions, equilibrium profiles from games with instant monitoring can be transferred to games with delayed monitoring.

▶ **Theorem 2.1.** *Let $\mathcal{G}$ be a game with instant monitoring and shift-invariant submixing payoffs, and let D be a finite delay space. Then, for every ergodic equilibrium payoff w in $\mathcal{G}$, there exists an equilibrium of the D-delayed monitoring game $\mathcal{G}'$ with the same payoff w.*

The proof relies on constructing a strategy for the delayed-monitoring game while maintaining a collection of virtual plays of the instant-monitoring game on which the given strategy is queried. The responses are then combined according to a specific schedule to ensure that the actual play arises as a shuffle of the virtual plays.

3 Proof

Consider a game $\mathcal{G} = (G, u)$ with instant monitoring where the payoff aggregation function u is shift-invariant and submixing, and suppose that $\mathcal{G}$ admits an equilibrium profile s. For an arbitrary finite delay space D, let $\mathcal{G}'$ be the delayed-monitoring variant of $\mathcal{G}$. In the following steps, we will construct a strategy profile s' for $\mathcal{G}'$, that is in equilibrium and yields the same payoff $u(s')$ as s in $\mathcal{G}$.

3.1 Unravelling small cycles

To minimise the combinatorial overhead for scheduling delayed responses, it is convenient to ensure that, whenever the play returns to a state v, the signals emitted at the previous visit at v have been received by all players. If every cycle in the given game graph G is at least as long as any possible delay, this is clearly satisfied. Otherwise, the graph can be expanded to avoid small cycles, e.g., by taking the product with a cyclic group of order equal to the maximal delay.

Concretely, let m be the greatest delay among $\max D^i$, for all players i. We define a new game graph $\hat{G}$ as the product of G with the additive group $\mathbb{Z}_m$ of integers modulo m, over the state set $\{v_j \mid v \in V, j \in \mathbb{Z}_m\}$ by allowing transitions (v_j, a, b, v'_{j+1}), for every $(v, a, b, v') \in E$ and all $j \in \mathbb{Z}_m$, and by assigning stage payoffs $\hat{p}^i(v_j, a, b, v'_{j+1}) := p^i(v, a, b, v')$, for all transitions $(v, a, b, v') \in E$. Obviously, every cycle in this game has length at least m. Moreover, the games $(\hat{G}, u)$ and (G, u) are equivalent: Since the index component $j \in \mathbb{Z}_m$ is not observable to the players, the two games have the same sets of strategies, and profiles of corresponding strategies yield the same observable play outcome, and hence the same payoffs.

In conclusion, we can assume without loss of generality that each cycle in the game graph G is longer than the maximal delay $\max D^i$, for all players i.

3.2 The Frankenstein procedure

We describe a strategy f^i for Player i in the delayed monitoring game G' by a reactive procedure that receives observations of states and signals as input and produces actions as output.

The procedure maintains a collection of virtual plays of the instant-monitoring game. More precisely, these are observation histories for Player i following the strategy s^i in G, which we call *threads*. The observations collected in a thread $\pi = v_0,\ a_1^i, (b_1^i, d_1^i), v_1,\ \ldots,\ a_r^i, (b_r^i, d_r^i), v_r$ are drawn from the play of the main delayed-monitoring game G'. Due to delays, it may occur that the signal (b_r^i, d_r^i) emitted in the last period of a thread has not yet been received. In this case, the signal entry is replaced by a special symbol #, and we say that the thread is *pending*. As soon as the player receives the signal, the placeholder # is overwritten with the actual value, and the thread becomes *active*. Active threads π are used to query the strategy s^i; the prescribed action $a^i = s^i(\pi)$ is played in the main delayed-monitoring game and it is also used to continue the thread of the virtual instant-monitoring game.

To be continued, a thread must be active and its current state needs to match the actual state of the play in the delayed-monitoring game. Intuitively, threads advance more slowly than the actual play, so we need multiple threads to keep pace with it. Here, we use a collection of $|V| + 1$ threads, indexed by an ordered set $K = V \cup \{\varepsilon\}$. The main task of the procedure is to schedule the continuation of threads. To do so, it maintains a data structure (τ, h) that consists of the threads $\tau = (\tau_k)_{k \in K}$ and a scheduling sequence $h = h[0], \dots, h[t]$ of indices from K, at every period $t \geq 0$ of the actual play. For each previous $r < t$, the entry $h[r]$ points to the thread according to which the action of period $r + 1$ in the actual play has been prescribed; the last entry $h[t]$ points to an active thread that is currently scheduled for prescribing the action to be played next.

The version of Procedure Frankenstein^i for Player i, given below, is parametrised by the game graph G with the designated initial state, the delay space D^i, and the given equilibrium strategy s^i in the instant-monitoring game. In the initialisation phase, the initial state v_0 is stored in the initial thread τ_ε to which the current scheduling entry $h[0]$ points. The remaining threads are initialised, each with a different position from V. Then, the procedure enters a non-terminating loop along the periods of the actual play. In every period t, it outputs the action prescribed by strategy s^i for the current thread scheduled by $h[t]$ (Line 5). Upon receiving the new state, this current thread is updated by recording the played action and the successor state; as the signal emitted in the instant-monitoring play is not available in the delayed-monitoring variant, it is temporarility replaced by #, which marks the current thread as pending (Line 7). Next, an active thread that matches the new state is scheduled (Line 9), and the received signals are recorded with the pending threads to which they belong (Line 11 – 14). As a consequence, these threads become active.

3.3 Correctness

In the following, we argue that the procedure Frankenstein^i never violates the assertions in Line 4, 8, and 13 while interacting with Nature in the delayed-monitoring game $\mathcal{G}'$, and thus implements a valid strategy for Player i.

Specifically, we show that for every history

$$\pi = v_0,\ a_1, (b_1, d_1), v_1,\ \dots,\ a_t, (b_t, d_t), v_t$$

in the delayed-monitoring game that follows the prescriptions of the procedure up to period $t > 0$, (1) the scheduling function $h[t] = k$ points to an active thread τ_k that ends at state v_t, and (2) for the state v_{t+1} reached by playing $a_{t+1} := s^i(\tau_k)$ at π, there exists an active thread $\tau_{k'}$ that ends at v_{t+1}. We proceed by induction over the period t. In the base case, both properties hold, due to the way in which the data structure is initialised: the (trivial) thread τ_ε is active, and for any successor state v_1 reached by $a_1 := s^i(\tau_\varepsilon)$, there is a fresh thread τ_{v_1} that is active. For the induction step in period $t + 1$, property (1) follows from property (2) of period t. To verify that property (2) holds, we distinguish two cases. If v_{t+1} did not occur previously in π, the initial thread $\tau_{v_{t+1}}$ still consists of the trivial history v_{t+1}, and it is thus active. Else, let $r < t$ be the period in which v_{t+1} occurred last. Then, for $k' = h[r]$, the thread $\tau_{k'}$ ends at v_{t+1}. Moreover, by our assumption that the cycles in G are longer than any possible delay, it follows that the signals emitted in period $r < t - m$ have been received along π and were recorded (Line 12–14). Hence, $\tau_{k'}$ is an active thread ending at v_{t+1}, as required.

```
Procedure: Frankenstein^i(G, v_0, D^i, s^i)
   // initialisation
1  τ_ε := v_0; h[0] = ε
2  foreach v ∈ V do τ_v := v

   // play loop
   for t = 0 to ω do
3      k := h[t]
4      assert (τ_k is an active thread)
5      play action a^i := s^i(τ_k)                       // a^i_{t+1}
6      receive new state v                               // v_{t+1}
7      update τ_k := τ_k a^i#v
8      assert (there exists an index k' ≠ k such that τ_{k'} ends at state v)
9      set h[t+1] to the least such index k'
10     receive observation z^i ⊆ B^i × D^i                    // z^i_{t+1}
11     foreach (b^i, d^i) ∈ z^i do
12         k := h[t − d^i]
13         assert (τ_k = ρ#v', for some prefix ρ, state v')
14         update τ_k := ρ(b^i, d^i)v'
       end
   end
```

To see that the assertion of Line 13 is never violated, we note that every observation history $\beta^i(\pi)$ of the actual play π in $\mathcal{G}'$ up to period t corresponds to a finitary shuffle of the threads τ in the t-th iteration of the play loop, described by the scheduling function h: The observations $(a^i_r, (b_r, d_r)^i, v_r)$ associated to any period $r \leq t$ appear at the end of $\tau_{h[r]}$, if the signal $(b_r, d_r)^i$ was delivered until period t, and with the placeholder #, otherwise.

In summary, it follows that the reactive procedure Frankenstein^i never halts, and it returns an action for every observed history $\beta^i(\pi)$ associated to an actual history π that follows it. Thus, the procedure defines a strategy $f^i : V(A^i 2^{Y^i} V)^* \to A^i$ for Player i.

3.4 Equilibrium condition

Finally, we show that the interplay of the strategies f^i described by the reactive procedure Frankenstein^i, for each player i, constitutes an equilibrium profile for the delayed-monitoring game $\mathcal{G}'$ yielding the same payoff as s in $\mathcal{G}$.

According to our remark in the previous subsection, every transition taken in a play π that follows the strategy f^i in $\mathcal{G}'$ is also observed in some thread history, which in turn follows s^i. Along the non-terminating execution of the reactive Frankenstein^i procedure, some threads must be scheduled infinitely often, and thus correspond to observations of plays in the perfect-monitoring game G. We argue that the observation by Player i of a play that follows the strategy f^i corresponds to a shuffle of such infinite threads (after discarding finite prefixes).

To make this more precise, let us fix a play π that follows f^i in $\mathcal{G}'$, and consider the infinite scheduling sequence $h[0], h[1], \ldots$ generated by the procedure. Since there are finitely many thread indices, some must appear infinitely often in this sequence; we denote by $L^i \subseteq K$ the subset of these indices, and look at the least period ℓ^i, after which only threads in L^i are scheduled. Then, the suffix of the observation $\beta^i(\pi)$ from period ℓ^i onwards can be written as a $|L^i|$-partite shuffle of suffixes of the threads τ_k for $k \in L^i$.

By our assumption that the payoff aggregation function u is shift-invariant and submixing, it follows that the payoff $u^i(\pi)$ lies between $\min\{u^i(\tau_k) \mid k \in L^i\}$ and $\max\{u^i(\tau_k) \mid k \in L^i\}$. Now, we apply this reasoning to all players to show that f^i is an equilibrium profile with payoff $u(s)$.

To see that the profile f in the delayed-monitoring game $\mathcal{G}'$ yields the same payoff as s in the instant-monitoring game $\mathcal{G}$, consider the unique play π that follows f, and construct L^i, for all players i, as above. Then, all threads of all players i follow s^i, which by ergodicity implies, for each infinite thread τ_k with $k \in L^i$ that $u^i(\tau_k) = u^i(s)$. Hence $\min\{u^i(\tau_k) \mid k \in L^i\} = \max\{u^i(\tau_k) \mid k \in L^i\}= u^i(\pi)$, for each player i, and therefore $u(f) = u(s)$.

To verify that f is indeed an equilibrium profile, consider a strategy g^i for the delayed-monitoring game and look at the unique play π that follows (f^{-i}, g^i) in $\mathcal{G}'$. Towards a contradiction, assume that $u^i(\pi) > u^i(f)$. Since $u^i(\pi) < \max\{u^i(\tau_k) \mid k \in L^i\}$, there must exist an infinite thread τ_k with index $k \in L^i$ such that $u^i(\tau_k) > u^i(f) = u^i(s)$. But τ_k corresponds to the observation $\beta^i(\rho)$ of a play ρ that follows s^{-i} in $\mathcal{G}$, and since s is an equilibrium strategy we obtain $u^i(s) \geq u^i(\rho) = u^i(\tau_k)$, a contradiction. This concludes the proof of our theorem.

3.5 Finite-state strategies

The transfer theorem makes no assumption on the complexity of equilibrium strategies in the instant-monitoring game at the outset; informally, we may think of these strategies as oracles that the Frankenstein procedure can query. Moreover, the procedure itself runs for infinite time along the periods the play, and the data structure it maintains grows unboundedly.

However, if we set out with an equilibrium profile of finite-state strategies, it is straightforward to rewrite the Frankenstein procedure as a finite-state automaton: instead of storing the full histories of threads, it is sufficient to maintain the current state reached by the strategy automaton for the relevant player after reading this history, over a period that is sufficiently long to cover all possible delays.

▶ **Corollary 3.1.** *Let $\mathcal{G}$ be a game with instant monitoring and shift-invariant submixing payoffs, and let D be a finite delay space. Then, for every ergodic payoff w in $\mathcal{G}$ generated by a profile of finite-state strategies, there exists an equilibrium of the D-delayed monitoring game $\mathcal{G}'$ with the same payoff w that is also generated by a profile of finite-state strategies.*

4 Conclusion

We presented a transfer result that implies effective solvability of concurrent games with a particular kind of imperfect information, due to imperfect monitoring of actions, and delayed delivery of signals. This is a setting where we cannot rely on grim-trigger strategies, typically used for constructing Nash equilibria in games of infinite duration for automated verification. Our method overcomes this obstacle by adapting the idea of delayed-response strategies of [8] from infinitely repeated games, with one state, to arbitrary finite state-transition structures.

Our transfer result imposes stronger restrictions than the one in [8], in particular, it does not cover discounted payoff functions. Nevertheless, the class of submixing payoff functions is general enough to cover most applications relevant in automated verification and synthesis.

The restriction to ergodic payoffs was made for technical convenience. We believe it is not critical for using the main result: The state space of every game can be partitioned into ergodic regions, where all initial states lead to the same equilibrium value. As the outcome of every equilibrium profile will stay within an ergodic region, we may analyse each ergodic region in separation, and apply standard zero-sum techniques to combine the results. A challenging open question is whether the assumption of perfect information about the current state can be relaxed.

Acknowledgement. This work was supported by the Indo-French Formal Methods Lab (LIA Informel) and by the European Union Seventh Framework Programme under Grant Agreement 601148 (CASSTING).

References

1 Salman Azhar, Gary Peterson, and John Reif. Lower bounds for multiplayer non-cooperative games of incomplete information. *Journal of Computers and Mathematics with Applications*, 41:957–992, 2001.

2 Dietmar Berwanger, Łukasz Kaiser, and Bernd Puchala. Perfect-information construction for coordination in games. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011), Proc.*, volume 13 of *Leibniz International Proceedings in Informatics*, pages 387–398, Mumbai, India, December 2011. Leibniz-Zentrum für Informatik.

3 Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Nash equilibria in concurrent games with Büchi objectives. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011), Proc.*, volume 13 of *LIPIcs*, pages 375–386. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2011.

4 Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Pure Nash equilibria in concurrent games. *Logical Methods in Computer Science*, 11(2:9), June 2015.

5 Thomas Brihaye, Véronique Bruyére, and Julie De Pril. On equilibria in quantitative games with reachability/safety objectives. *Theory of Computing Systems*, 54(2):150–189, 2014.

6 Krishnendu Chatterjee, Laurent Doyen, Emmanuel Filiot, and Jean-François Raskin. Doomsday equilibria for omega-regular games. In KennethL. McMillan and Xavier Rival, editors, *Verification, Model Checking, and Abstract Interpretation*, volume 8318 of *Lecture Notes in Computer Science*, pages 78–97. Springer Berlin Heidelberg, 2014.

7 B. Finkbeiner and S. Schewe. Uniform distributed synthesis. In *Logic in Computer Science (LICS'05), Proc.*, pages 321–330. IEEE, 2005.

8 Drew Fudenberg, Yuhta Ishii, and Scott Duke Kominers. Delayed-response strategies in repeated games with observation lags. *J. Economic Theory*, 150:487–514, 2014.

9 Hugo Gimbert and Edon Kelmendi. Two-player perfect-information shift-invariant submixing stochastic games are half-positional. *CoRR*, abs/1401.6575, 2014.

10 Orna Kupferman and Moshe Y. Vardi. Synthesizing distributed systems. In *Logic in Computer Science (LICS'01), Proc.*, pages 389–398. IEEE Computer Society Press, June 2001.

11 Gary L. Peterson and John H. Reif. Multiple-Person Alternation. In *Proc 20th Annual Symposium on Foundations of Computer Science, (FOCS 1979)*, pages 348–363. IEEE, 1979.

12 Amir Pnueli and Roni Rosner. Distributed reactive systems are hard to synthesize. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science, (FoCS '90)*, pages 746–757. IEEE Computer Society Press, 1990.

13 Dinah Rosenberg, Eilon Solan, and Nicolas Vieille. Stochastic games with imperfect monitoring. In Alain Haurie, Shigeo Muto, LeonA. Petrosjan, and T.E.S. Raghavan, editors, *Advances in Dynamic Games*, volume 8 of *Annals of the International Society of Dynamic Games*, pages 3–22. Birkhäuser Boston, 2006.

14 Eran Shmaya. The determinacy of infinite games with eventual perfect monitoring. *Proc. Am. Math. Soc.*, 139(10):3665–3678, 2011.

15 Michael Ummels. The complexity of Nash equilibria in infinite multiplayer games. In Roberto Amadio, editor, *Foundations of Software Science and Computational Structures*, volume 4962 of *Lecture Notes in Computer Science*, pages 20–34. Springer Berlin Heidelberg, 2008.

16 Michael Ummels and Dominik Wojtczak. The complexity of Nash equilibria in limit-average games. In Joost-Pieter Katoen and Barbara König, editors, *CONCUR 2011 – Concurrency Theory*, volume 6901 of *Lecture Notes in Computer Science*, pages 482–496. Springer Berlin Heidelberg, 2011.

17 Michael Ummels and Dominik Wojtczak. The complexity of Nash equilibria in stochastic multiplayer games. *Logical Methods in Computer Science*, 7(3), 2011.