Solving Concurrent Games in PSPACE

Patrick Totzke
Concurrent Games

á la Shapley [S’53], Everett [E’57], Gillette [G’57]:

- Played on a finite directed graph;
- time is discrete and not bounded;
- finite action sets;

At any time, players simultaneously chose an action each which, together with the current vertex, determines where to move next and a payoff vector.

\[ \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{r_n} P_n \]

probability space over infinite paths and expected payoffs.
Concurrent Games

á la Shapley [S’53], Everett [E’57], Gillette [G’57]:

- Played on a finite directed graph;
- time is discrete and not bounded;
- finite action sets;

At any time, players simultaneously chose an action each which, together with the current vertex, determines where to move next and a payoff vector.

\[ \Rightarrow \text{probability space over infinite paths and expected payoffs.} \]

Objectives: discounted sum; reachability; parity; mean-payoff
Concurrent Games

á la Shapley [S’53], Everett [E’57], Gillette [G’57]:

- Played on a finite directed graph;
- Time is discrete and not bounded;
- Finite action sets;

At any time, players simultaneously choose an action each which, together with the current vertex, determines where to move next and a payoff vector.

\[ \sim \] probability space over infinite paths and expected payoffs.

Objectives: discounted sum; reachability; parity; mean-payoff
Concurrent Games

á la Shapley [S’53], Everett [E’57], Gillette [G’57]:

▶ Played on a finite directed graph;
▶ time is discrete and not bounded;
▶ finite action sets;

At any time, players simultaneously chose an action each which, together with the current vertex, determines where to move next and a payoff vector.

⇝ probability space over infinite paths and expected payoffs.

Objectives: discounted sum; reachability; parity; mean-payoff
Concurrent Games

á la Shapley [S’53], Everett [E’57], Gillette [G’57]:

- Played on a finite directed graph;
- time is discrete and not bounded;
- finite action sets;

At any time, players simultaneously chose an action each which, together with the current vertex, determines where to move next and a payoff vector.

\[ \sim \text{probability space over infinite paths and expected payoffs.} \]

Objectives: discounted sum; reachability; parity; mean-payoff
Concurrent Games

á la Shapley [S’53], Everett [E’57], Gillette [G’57]:

▶ Played on a finite directed graph;
▶ time is discrete and not bounded;
▶ finite action sets;

At any time, players simultaneously chose an action each which, together with the current vertex, determines where to move next and a payoff vector.

⇝ probability space over infinite paths and expected payoffs.

Objectives: discounted sum; reachability; parity; mean-payoff $= \lim_{n \to \infty} \inf \sum_{i=1}^{n} r_n / n$
Concurrent Reachability Games

Example: Snowball game [dAHK’98]

- two players simultaneously chose actions (T/W, R/H);
- P1 wins on T+H or W+R, P2 wins on T+R, else repeat

In general,

- zero-sum values exist but optimal strategies may not
- values may not be rational, so can only be approximated [S’53]
- $\varepsilon$-optimal stationary and rational-valued strategies exist [E’57] but require doubly-exponentially small probabilities ($\varepsilon^2 O(N)$) [HKM’09]

- games can be solved by reduction to $\text{FO}(R)$, value or strategy iteration, in $\text{EXP}$. 

Patrick Totzke (Liverpool)
Concurrent Reachability Games

Example: Snowball game [dAHK’98]

- two players simultaneously chose actions (T/W, R/H);
- P1 wins on T+H or W+R, P2 wins on T+R, else repeat

In general,
- zero-sum values exist but optimal strategies may not
- values may not be rational, so can only be approximated [S’53]
- $\varepsilon$-optimal stationary and rational-valued strategies exist [E’57]
  but require doubly-exponentially small probabilities ($\varepsilon^2 O(N)$)[HKM’09]
- games can be solved by reduction to FO(R), value or strategy iteration, in EXP.
Concurrent Reachability Games

Example: Snowball game [dAHK’98]

- two players simultaneously chose actions (T/W, R/H);
- P1 wins on T+H or W+R, P2 wins on T+R, else repeat

In general,

- zero-sum values exist but optimal strategies may not
- values may not be rational, so can only be approximated [S’53]
- $\varepsilon$-optimal stationary and rational-valued strategies exist [E’57]
  but require doubly-exponentially small probabilities ($\varepsilon^{2^{\Omega(N)}}$)[HKM’09]
- games can be solved by reduction to $\text{FO}(\mathbb{R})$, value or strategy iteration, in EXP.
Concurrent Reachability Games

Example: Snowball game [dAHK’98]

- two players simultaneously chose actions (T/W, R/H);
- P1 wins on T+H or W+R, P2 wins on T+R, else repeat

In general,

- zero-sum values exist but optimal strategies may not
- values may not be rational, so can only be approximated [S’53]
- $\varepsilon$-optimal stationary and rational-valued strategies exist [E’57] but require doubly-exponentially small probabilities ($\varepsilon^{2^\Omega(N)}$)[HKM’09]
- games can be solved by reduction to $\text{FO}(\mathbb{R})$, value or strategy iteration, in EXP.
Solving Concurrent Reachability Games in $\text{FNP}^\text{NP}$

Approximating the Value of a Concurrent Reachability Game in the Polynomial Time Hierarchy

Sören Kristoffer Stil Frederiksen and Peter Bro Miltersen
Aarhus University

Abstract. We show that the value of a finite-state concurrent reachability game can be approximated to arbitrary precision in $\text{FNP}^\text{NP}$, that is, in the polynomial time hierarchy. Previously, no better bound than $\text{TFNP}$ was known for this problem. The proof is based on reformulating a variant of the state reduction algorithm for Markov chains using arbitrary precision floating point arithmetic and giving a rigorous error analysis of the algorithm.

1 Introduction

A concurrent reachability game (e.g., $\text{REACH}_\text{NP}$) $G$ is a finitely presented two-player game of potentially infinite duration, played between Player 1, the reachability player, and Player 2, the safety player. The arena of the game consists of a finite set of positions $0, 1, 2, \ldots, N$. When play begins, a pebble rests at position 1, the “start position”. At each stage of play, with the pebble resting at a particular “current” position $i$, Player 1 chooses an action $i \in \{1, 2, \ldots, m\}$ while Player 2 chooses an action $j \in \{1, 2, \ldots, n\}$. A fixed and commonly known transition function $\pi : \{1, 2, \ldots, N\} \times \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\} \to \{0, 1, 2, \ldots, N\}$ determines the next position of the pebble, namely $\pi(k, i, j)$. If the pebble ever reaches $0$ (the “goal position”), play ends, and Player 1 wins the game. If the pebble never reaches goal, Player 2 wins.

A stationary strategy for a player is a family of probability distribution on his available moves at each state of the game. Probst showed that every concurrent

1. Guess a strategy profile (and value) in floating point notations $x \cdot 10^y$ with $x, y \leq 2^\mathcal{O}(n)$

2. Verify that the resulting Markov chain satisfies the reachability probability
Solving Concurrent Reachability Games in $\text{FNP}^{\text{NP}}$

1. Guess a strategy profile (and value) in floating point notations $x \cdot 10^y$ with $x, y \leq 2^{O(n)}$
   - requires “approximately normalised” representations of distributions
2. Verify that the resulting Markov chain satisfies the reachability probability
Solving Concurrent Reachability Games in $\text{FNP}^\text{NP}$

1. Guess a strategy profile (and value) in floating point notations $x \cdot 10^y$ with $x, y \leq 2^{O(n)}$
   - requires “approximately normalised” representations of distributions

2. Verify that the resulting Markov chain satisfies the reachability probability
   - state-space reduction uses few operations, each introduces a bounded error.
Solving Concurrent Reachability Games in $\text{FNP}^{\text{NP}}$

1. Guess a strategy profile (and value) in floating point notations $x \cdot 10^y$ with $x, y \leq 2^{O(n)}$
   - requires “approximately normalised” representations of distributions

2. Verify that the resulting Markov chain satisfies the reachability probability
   - state-space reduction uses few operations, each introduces a bounded error.
Concurrent Parity Games

Example: The Bad Match [T’92, MS’96]

- two players simultaneously chose 0 or 1;
- if P1 picks 1 both choices are fixed forever
- (only) matches are accepting;

![Parity Game Diagram]

Patrick Totzke (Liverpool)
Concurrent Parity Games

Example: The Bad Match [T’92, MS’96]

- two players simultaneously chose 0 or 1;
- if P1 picks 1 both choices are fixed forever
- (only) matches are accepting;

Value(d)=1. No optimal strategy exists and no FM or Markov strategy is $\varepsilon$-optimal.
**Concurrent Parity Games**

**Example: The Bad Match [T’92, MS’96]**

- two players simultaneously chose 0 or 1;
- if P1 picks 1 both choices are fixed forever
- (only) matches are accepting;

Value(d) = 1. No optimal strategy exists and no FM or Markov strategy is $\varepsilon$-optimal.

- Markov+1bit $\varepsilon$-optimal strategies always exists for Büchi Games [KMST’24]
- Parity games reduce to stateful-discounted games [dAHM’03, GZ’06], which can be solved in $\text{FNP}^{\text{NP}}$ [ACSS’24].
Concurrent Mean-Payoff Games

Example: The Big Match [BF’68]

- two players simultaneously chose 0 or 1;
- if P2 picks 1 both choices are fixed forever;
- P1 gains 1 on a match, 0 otherwise.

In general,
- zero-sum values exist but may be irrational [S’53]
- optimal strategies and NE need not exist! [E’57]
- stationary $\varepsilon$-optimal need not exist! [BF’68]
- open: do $\varepsilon$-NE always exist?
Concurrent Mean-Payoff Games

Example: The Big Match [BF’68]

- two players simultaneously chose 0 or 1;
- if P2 picks 1 both choices are fixed forever;
- P1 gains 1 on a match, 0 otherwise.

In general,

- zero-sum values exist but may be irrational [S’53]
- optimal strategies and NE need not exist! [E’57]
- stationary $\varepsilon$-optimal need not exist! [BF’68]
Status Quo: Solving Concurrent MPG

- Mertens & Neymann ’89: Value iteration $\varepsilon^{-mO(n)}$
- Hansen, Koucjy, Lauritzen, Miltersen, Tsigaridas’11: Recursive binary search $\log(\varepsilon^{-1})m^{2O(n)}$
- Oliu-Barton ’20: Binary search $\log(\varepsilon^{-1})m^{O(n)}$
- Hansen, Koucky, Lauritzen, Miltersen’11: For any discount factor $\varepsilon^{mO(n)}$ the discounted game approximates the value of the mean-payoff game within $\varepsilon$.
  Etessami, Yannakakis ’05 lets us find the value of a discounted game in $\exists FO(\mathbb{R})$
Finding good Bounded Strategies

Definition

A strategy $\epsilon$-achieves value $v \in \mathbb{R}$ with respect to $S$ if it guarantees expected outcome of at least $v - \epsilon$ against all opponent strategies in $S$.

Let’s consider bounded-memory strategies.

Theorem

The following is in FNP$^{NP}$. Given a game, bounds $b \in \mathbb{N}^k$, $\epsilon \in (0, 1/2)$ and $v \in \mathbb{R}$, find a strategy that $\epsilon$-achieves $v$ with respect to $B(b)$ if for some $\epsilon' \in [0, \epsilon/3)$, some strategy $\epsilon'$-achieves $v$ with respect to $B(b)$.

We assume $b$ is unary, $\epsilon$ is binary, and transition matrices are in ANR floating points.
Proof Ideas

We follow the approach of [FM’13] as much as possible.

Part 1 (guess profiles):
Extra Lemma (**): If good profiles exist, then also one in ANR.

Part 2 (approximate values in MCs): via state-reductions
Proof Ideas

We follow the approach of [FM’13] as much as possible.

Part 1 (guess profiles):
Extra Lemma (**): If good profiles exist, then also one in ANR.

Part 2 (approximate values in MCs): via state-reductions that preserve probabilities, rewards and durations.
Proof Ideas

We follow the approach of [FM’13] as much as possible.

Part 1 (guess profiles):
Extra Lemma (**): If good profiles exist, then also one in ANR.

Part 2 (approximate values in MCs): via state-reductions that preserve probabilities, rewards and durations.

and therefore the “mpo-ratio” \( \liminf_{n \to \infty} \frac{\sum_{i=1}^{n} r_i}{\sum_{i=1}^{n} d_i} \)
Proof Ideas

We follow the approach of [FM’13] as much as possible.

Part 1 (guess profiles):
Extra Lemma (**) : If good profiles exist, then also one in ANR.

Part 2 (approximate values in MCs) : via state-reductions that preserve probabilities, rewards and durations.

and therefore the “mpo-ratio” \( \liminf_{n \to \infty} \frac{\sum_{i=1}^{n} r_i}{\sum_{i=1}^{n} d_i} = \text{mpv}. \)
Proof Ideas

We follow the approach of [FM’13] as much as possible.

Part 1 (guess profiles):
Extra Lemma (**): If good profiles exist, then also one in ANR.

Part 2 (approximate values in MCs): via state-reductions that preserve probabilities, rewards and durations.

\[
\text{and therefore the "mpo-ratio" } \liminf_{n \to \infty} \sum_{i=1}^{n} r_i / \sum_{i=1}^{n} d_i = \text{mpv}.
\]

This can be encoded in \( \text{FO}(\mathbb{R}) \), + perturbation lemma gives (**).
This works for any fixed number $k$ of players with individual MP objectives and partial information!

Implications

- zero-sum values in (perfect-info) MPG can be approximated, existence of memoryless $(c_{-})NE$ can be checked in FNP[NP].
- results apply to parity with minimal adjustments in SSRs
- Stay-in-a-set games
- Quitting games

---

**Bounded Memory Strategies in Partial-Information Games**

Sougata Jaiswal
University of Liverpool, UK

Patrick Totzke
University of Liverpool, UK

**ABSTRACT**

We study the computational complexity of solving stochastic games with a fixed number of players. We consider classes of games where players play in rounds and receive feedback at the end of each round. The feedback consists of a vector of partial information for each player, which is a subset of their private state. The goal is to find a strategy that maximizes the expected total reward over a fixed number of rounds.

We classify the complexity of finding optimal strategies based on the amount of information available to each player. The classes range from full information to no information at all. We show that for games with perfect information, optimal strategies can be found in polynomial time. For games with partial information, the problem becomes more complex, and we provide hardness results for different classes of games.

**Keywords**

- Stochastic games
- Partial information
- Memoryless strategies
- Algorithmic complexity

**INTRODUCTION**

In recent years, there has been significant interest in the study of partial-information games, which have applications in various fields such as economics, computer science, and operations research. A partial-information game is a stochastic game where players have limited information about the state of the game at each round. The amount of information available to each player is modeled by a vector of partial information, which is a subset of their private state.

The goal of a player in a partial-information game is to find a strategy that maximizes their expected total reward over a fixed number of rounds. This problem is known to be computationally challenging, and the complexity depends on the amount of information available to each player.

In this paper, we focus on the computational complexity of finding optimal strategies for partial-information games. We classify the complexity based on the amount of information available to each player and provide hardness results for different classes of games.