Algorithms for Stochastic Games

Hugo Gimbert, CNRS, LaBRI, Bordeaux

March 3, 2009
Algorithms and Games

Solving Simple Stochastic Games

Solving Stochastic Games

Solving Stochastic Games with Signals

Conclusion
Computable functions

- Computability and algorithms.
Computable functions

- Computability and algorithms.
- **Definition**: a function $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ is **computable** if it is computable by a Turing machine. Equivalent to Pascal programs which terminate. Alphabet $\{0, 1\}$ or $\Sigma$ finite.
Computable functions

- Computability and algorithms.

- **Definition:** A function $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ is computable if it is computable by a Turing machine. Equivalent to Pascal programs which terminate. Alphabet $\{0, 1\}$ or $\Sigma$ finite.

- Finitely many instructions. Un**limited** amount of memory. Computation in finite time.
Computable functions

- **Computability and algorithms.**
- **Definition:** a function \( f : \{0, 1\}^* \to \{0, 1\}^* \) is **computable** if it is computable by a Turing machine. Equivalent to Pascal programs which terminate. Alphabet \( \{0, 1\} \) or \( \Sigma \) finite.
- **Finitely** many instructions. **Unlimited** amount of memory. Computation in **finite time**.
- **Decidable problem:** \( L \subseteq \{0, 1\}^* \) whose indicator function is computable.
Computable functions

- **Computability and algorithms.**
- **Definition:** a function $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ is *computable* if it is computable by a Turing machine. Equivalent to Pascal programs which terminate. Alphabet $\{0, 1\}$ or $\Sigma$ finite.
- **Finitely many instructions.** *Unlimited* amount of memory. Computation in *finite time*.
- **Decidable problem:** $L \subseteq \{0, 1\}^*$ whose indicator function is computable.
- **Decidable problems:** words on $\{0, 1\}$ with more 0’s than 1’s. Matrices with coefficients in $\mathbb{Q}$ and maximal rank.
Computable functions

- **Computability and algorithms.**

- **Definition:** a function \( f : \{0, 1\}^* \rightarrow \{0, 1\}^* \) is **computable** if it is computable by a **Turing machine**. Equivalent to Pascal programs which terminate. Alphabet \( \{0, 1\} \) or \( \Sigma \) finite.

- **Finitely** many instructions. **Unlimited** amount of memory. Computation in **finite time**.

- **Decidable problem:** \( L \subseteq \{0, 1\}^* \) whose indicator function is computable.

- **Decidable problems:** words on \( \{0, 1\} \) with more 0’s than 1’s. Matrices with coefficients in \( \mathbb{Q} \) and maximal rank.

- **Undecidable problems:** Halting problem. Post Correspondence Problem.
A Decidable Game Problem

- Games on finite trees [Zermelo, 1913].
A Decidable Game Problem

- Games on finite trees [Zermelo, 1913].
- Example: game of chess. Tree whose nodes are all possible boards. Some leaves are marked with "win". Root is the initial board. White (resp. Black) chooses a child on even (resp. odd) levels.
A Decidable Game Problem

- Games on finite trees [Zermelo, 1913].
- Example: game of chess. Tree whose nodes are all possible boards. Some leaves are marked with "win". Root is the initial board. White (resp. Black) chooses a child on even (resp. odd) levels.
- Decision problem: can player 1 force a "win"?
A Decidable Game Problem

- Games on finite trees [Zermelo, 1913].
- Example: game of chess. Tree whose nodes are all possible boards. Some leaves are marked with "win". Root is the initial board. White (resp. Black) chooses a child on even (resp. odd) levels.
- Decision problem: can player 1 force a "win"?
- Algorithm: dynamic programming.
Games on finite trees

- **Algorithm**: dynamic programming.
Games on finite trees

- **Algorithm**: dynamic programming.
- If a node controlled by player 1 has at least one child marked "win" then mark the node "win".

Games on graphs: the same algorithm works for games on graphs. States $S = S_1 \cup S_2$ controlled by player 1 or 2.
Games on finite trees

- **Algorithm**: dynamic programming.
- If a node controlled by player 1 has at least one child marked "win" then mark the node "win".
- If a node controlled by player 2 has all children marked "win" then mark the node "win".

Games on graphs: the same algorithm works for games on graphs. States $S = S_1 \cup S_2$ controlled by player 1 or 2.
Games on finite trees

- **Algorithm**: dynamic programming.
- If a node controlled by player 1 has at least one child marked "win" then mark the node "win".
- If a node controlled by player 2 has all children marked "win" then mark the node "win".
- Answer yes if the root is marked with "win".
Games on finite trees

- **Algorithm**: dynamic programming.
- If a node controlled by player 1 has at least one child marked "win" then mark the node "win".
- If a node controlled by player 2 has all children marked "win" then mark the node "win".
- Answer yes if the root is marked with "win".
- Strategy for player 1: choose a son marked "win".
Games on finite trees

- **Algorithm**: dynamic programming.
- If a node controlled by player 1 has at least one child marked "win" then mark the node "win".
- If a node controlled by player 2 has all children marked "win" then mark the node "win".
- Answer yes if the root is marked with "win".
- Strategy for player 1: choose a son marked "win".
- **Games on graphs**: the same algorithm works for games on graphs. States $S = S_1 \cup S_2$ controlled by player 1 or 2.
Algorithms and Games

Solving Simple Stochastic Games

Solving Stochastic Games

Solving Stochastic Games with Signals

Conclusion
Simple Stochastic Games

- **Simple Stochastic Games.** Games on graphs with stochastic transitions. States $S = S_2 \cup S_2 \cup S_R$ controlled by player 1, player 2 and nature. Target state $t \in S$. 
Simple Stochastic Games

- **Simple Stochastic Games.** Games on graphs with stochastic transitions. States $S = S_2 \cup S_2 \cup S_R$ controlled by player 1, player 2 and nature. Target state $t \in S$.

- **Example**
Simple Stochastic Games

Simple Stochastic Games. Games on graphs with stochastic transitions. States $S = S_2 \cup S_2 \cup S_R$ controlled by player 1, player 2 and nature. Target state $t \in S$.

Example

Strategy: $\sigma : S^* S_1 \rightarrow D(S)$. 
Simple Stochastic Games

- **Simple Stochastic Games.** Games on graphs with stochastic transitions. States $S = S_2 \cup S_2 \cup S_R$ controlled by player 1, player 2 and nature. Target state $t \in S$.

- **Example**

- **Strategy:** $\sigma : S^* S_1 \rightarrow D(S)$.

- **Decision problem:** does player 1 has a strategy $\sigma$ for winning with probability $> \frac{1}{2}$?
Linear program for the one-player case

- Only one player: \( S_1 = \emptyset \). \( S = S_1 \cup S_R \).
Linear program for the one-player case

- Only one player: \( S_1 = \emptyset \). \( S = S_1 \cup S_R \).
- Value: of state \( s = \) supremum probability \( v(s) \) s.t. player 1 can guarantee winning probability \( \geq v(s) \).
Linear program for the one-player case

- **Only one player:** $S_1 = \emptyset$. $S = S_1 \cup S_R$.
- **Value:** of state $s = \sup$ probability $v(s)$ s.t. player 1 can guarantee winning probability $\geq v(s)$.
- **Value of target state** $= 1$. 
Linear program for the one-player case

- **Only one player:** $S_1 = \emptyset$. $S = S_1 \cup S_R$.
- **Value:** of state $s = \sup$ probability $v(s)$ s.t. player 1 can guarantee winning probability $\geq v(s)$.
- **Value of target state** $= 1$.
- **Value of** $s \in S_1 = \max$ value of its successors.
Linear program for the one-player case

- **Only one player**: $S_1 = \emptyset$. $S = S_1 \cup S_R$.
- **Value**: of state $s = \text{supremum probability } v(s) \text{ s.t. player 1 can guarantee winning probability } \geq v(s)$.
- Value of target state $= 1$.
- Value of $s \in S_1 = \text{maximal value of its successors}$.
- Value of $s \in S_R = \text{average value of successors}$.
Linear program for the one-player case

- **Only one player:** $S_1 = \emptyset$. $S = S_1 \cup S_R$.
- **Value:** of state $s = \sup$ probability $v(s)$ s.t. player 1 can guarantee winning probability $\geq v(s)$.
- **Value of target state** $= 1$.
- **Value of** $s \in S_1 =$ maximal value of its successors.
- **Value of** $s \in S_R =$ average value of successors.
- **Linear program. Solvable in polynomial time.**
Linear program for the one-player case

- Only one player: $S_1 = \emptyset$. $S = S_1 \cup S_R$.
- Value: of state $s = \sup$ probability $v(s)$ s.t. player 1 can guarantee winning probability $\geq v(s)$.
- Value of target state = 1.
- Value of $s \in S_1 = \max$ value of its successors.
- Value of $s \in S_R = \text{average value of successors}$.
- Linear program. Solvable in polynomial time.
- Minimize $\sum_s v(s)$ with constraints:
  \[ s \in S, \quad 0 \leq v(s) \leq 1 \]
  \[ t \text{ target}, \quad v(t) = 1 \]
  \[ s \in S_1, \quad (s, u) \in E, \quad v(s) \geq v(u) \]
  \[ s \in S_R, \quad v(s) = \sum_{u \in S} p(s, u) \cdot v(u) \]
Strategy enumeration for the two-player case

- **Decision problem**: does player 1 have a strategy $\sigma$ for winning with probability more than 1/2?
Strategy enumeration for the two-player case

- **Decision problem**: does player 1 has a strategy $\sigma$ for winning with probability more than $1/2$?
- **Theorem [Gillette 57, Liggett, Lippman 69]**: stationary deterministic strategies $\sigma : S_1 \to S$ are enough.
Strategy enumeration for the two-player case

- **Decision problem**: does player 1 have a strategy $\sigma$ for winning with probability more than $1/2$?

- **Theorem [Gillette 57, Liggett, Lippman 69]**: stationary deterministic strategies $\sigma : S_1 \rightarrow S$ are enough.

- **Algorithm for two players**: enumerate all stationary deterministic strategies $\sigma : S_1 \rightarrow S$. 

- Exponentially many strategies $\sigma \in S$. Exponential time.

- "Guess" good strategy $\sigma : S_1 \rightarrow S$. Check in polynomial time.

- Decision problem in $\text{NP} \cap \text{co-NP}$. Polynomial?

- Polynomial cases. Trees. Fixed number of random vertices.

- Generalization: perfect-information payoff games with stationary deterministic optimal strategies.
Strategy enumeration for the two-player case

▶ Decision problem: does player 1 have a strategy $\sigma$ for winning with probability more than 1/2?

▶ Theorem [Gillette 57, Liggett, Lippman 69]: stationary deterministic strategies $\sigma : S_1 \rightarrow S$ are enough.

▶ Algorithm for two players: enumerate all stationary deterministic strategies $\sigma : S_1 \rightarrow S$.

▶ Exponentially many strategies $\sigma \in S^{S_1}$. Exponential time.
Strategy enumeration for the two-player case

- **Decision problem**: does player 1 has a strategy $\sigma$ for winning with probability more than 1/2?

- **Theorem [Gillette 57, Liggett, Lippman 69]**: stationary deterministic strategies $\sigma : S_1 \to S$ are enough.

- **Algorithm for two players**: enumerate all stationary deterministic strategies $\sigma : S_1 \to S$.

- **Exponentially many strategies** $\sigma \in S^{S_1}$. Exponential time.

- **”Guess” good strategy** $\sigma : S_1 \to S$. Check in polynomial time.
Strategy enumeration for the two-player case

- **Decision problem**: does player 1 has a strategy $\sigma$ for winning with probability more than $1/2$?

- **Theorem [Gillette 57, Liggett, Lippman 69]**: stationary deterministic strategies $\sigma : S_1 \to S$ are enough.

- **Algorithm for two players**: enumerate all stationary deterministic strategies $\sigma : S_1 \to S$.

- **Exponentially many strategies** $\sigma \in S^{S_1}$. Exponential time.

- **”Guess” good strategy** $\sigma : S_1 \to S$. Check in polynomial time.

- **Decision problem in $\text{NP} \cap \text{co-NP}$**: Polynomial?
Strategy enumeration for the two-player case

- **Decision problem**: does player 1 have a strategy $\sigma$ for winning with probability more than 1/2?
- **Theorem [Gillette 57, Liggett, Lippman 69]**: stationary deterministic strategies $\sigma : S_1 \to S$ are enough.
- **Algorithm for two players**: enumerate all stationary deterministic strategies $\sigma : S_1 \to S$.
- **Exponentially many strategies** $\sigma \in S^{S_1}$. Exponential time.
- **”Guess” good strategy** $\sigma : S_1 \to S$. Check in polynomial time.
- Decision problem in $\text{NP} \cap \text{co-NP}$. Polynomial?
- **Polynomial cases**: Trees. Fixed number of random vertices.
Strategy enumeration for the two-player case

- **Decision problem**: does player 1 have a strategy $\sigma$ for winning with probability more than $1/2$?
- **Theorem [Gillette 57, Liggett, Lippman 69]**: stationary deterministic strategies $\sigma : S_1 \rightarrow S$ are enough.
- **Algorithm for two players**: enumerate all stationary deterministic strategies $\sigma : S_1 \rightarrow S$.
- **Exponentially many strategies** $\sigma \in S^{S_1}$. Exponential time.
- **”Guess”** good strategy $\sigma : S_1 \rightarrow S$. Check in polynomial time.
- **Decision problem in $\text{NP} \cap \text{co-NP}$**: Polynomial?
- **Polynomial cases**: Trees. Fixed number of random vertices.
- **Generalization**: perfect-information payoff games with stationary deterministic optimal strategies.
Algorithms and Games

Solving Simple Stochastic Games

Solving Stochastic Games

Solving Stochastic Games with Signals

Conclusion
Stochastic Games

- **Stochastic Games** [Shapley 53]. States $S$, actions $I$ and $J$. Players **play simultaneously.** Rewards $x_0, x_1, \ldots$ player 1 receives $x_0 + \lambda x_1 + \lambda x_2 + \cdots$ from player 1.
Stochastic Games

- **Stochastic Games [Shapley 53].** States $S$ actions $I$ and $J$. Players play simultaneously. Rewards $x_0, x_1, \ldots$ player 1 receives $x_0 + \lambda x_1 + \lambda x_2 + \cdots$ from player 1.
- Example.
Stochastic Games

- **Stochastic Games** [Shapley 53]. States $S$ actions $I$ and $J$. Players *play simultaneously*. Rewards $x_0, x_1, \ldots$ player 1 receives $x_0 + \lambda x_1 + \lambda x_2 + \cdots$ from player 1.

- Example.

- **Decision problem**: does player 1 has a strategy for ensuring payoff $> 0$?
Stochastic Games

- **Stochastic Games [Shapley 53]**. States $S$, actions $I$ and $J$.
  Players play simultaneously. Rewards $x_0, x_1, \ldots$ player 1 receives $x_0 + \lambda x_1 + \lambda x_2 + \cdots$ from player 1.

- **Example**.

- **Decision problem**: does player 1 has a strategy for ensuring payoff $> 0$?

- **Shapley algorithm**. Compute value (optimal payoff) of the game in $0, 1, 2, 3, \ldots$ steps.
Stochastic Games

- **Stochastic Games [Shapley 53]**. States $S$, actions $I$ and $J$. Players play simultaneously. Rewards $x_0, x_1, \ldots$ player 1 receives $x_0 + \lambda x_1 + \lambda x_2 + \cdots$ from player 1.

- **Example**.

- **Decision problem**: does player 1 has a strategy for ensuring payoff $> 0$?

- **Shapley algorithm**. Compute value (optimal payoff) of the game in $0, 1, 2, 3, \ldots$ steps.

- **Strategy Improvement Algorithm** [Hoffman, Karp, 66] [Rao, Chandrasekaran and Nair, 73]. Compute better and better strategies.
Stochastic Games

- **Stochastic Games [Shapley 53]**. States $S$ actions $I$ and $J$. Players play simultaneously. Rewards $x_0, x_1, \ldots$ player 1 receives $x_0 + \lambda x_1 + \lambda x_2 + \cdots$ from player 1.

- **Example**.

- **Decision problem**: does player 1 has a strategy for ensuring payoff > 0?

- **Shapley algorithm**. Compute value (optimal payoff) of the game in $0, 1, 2, 3, \ldots$ steps.

- **Strategy Improvement Algorithm** [Hoffman, Karp, 66] [Rao, Chandrasekaran and Nair, 73]. Compute better and better strategies.

- **Not exact computation**: converge to the value but no guarantee on the number of steps for a given precision. May be efficient in practice.
Using first order theory on reals

- **Decision problem**: does player 1 have a strategy which guarantees payoff $> 0$?
Using first order theory on reals

- **Decision problem**: does player 1 have a strategy which guarantees payoff $> 0$?

- **Theorem [Shapley 53]**: it is enough to consider stationary (but not deterministic) strategies $\sigma : S \rightarrow D(I)$.
Using first order theory on reals

- **Decision problem**: does player 1 has a strategy which guarantees payoff $> 0$?
- **Theorem [Shapley 53]**: it is enough to consider stationary (but not deterministic) strategies $\sigma : S \rightarrow D(I)$.
- **First order theory on reals**: well-formed formula with rational constants $\frac{a}{b}$, arithmetic operations $\ast$ and $+$, variables $x_1, x_2, \ldots, x_n$, comparison $\leq$, quantifiers $\exists$ and $\forall$, boolean operators $\neg, \land, \lor$ and parentheses ( and ).

$\exists x, 3x^6 - 4x^2 + 3 = 0.$
Using first order theory on reals

- **Decision problem**: does player 1 have a strategy which guarantees payoff $> 0$?
- **Theorem [Shapley 53]**: it is enough to consider stationary (but not deterministic) strategies $\sigma : S \rightarrow D(I)$.
- **First order theory on reals**: well-formed formula with rational constants $\frac{a}{b}$, arithmetic operations $\ast$ and $+$, variables $x_1, x_2, \ldots, x_n$, comparison $\leq$, quantifiers $\exists$ and $\forall$, boolean operators $\neg, \land, \lor$ and parentheses ( and ). $\exists x, \frac{3x^6}{x^6} - 4x^2 + 3 = 0$.
- **Theorem [Tarski, 51]**: quantifier elimination. Truth of first order formula on reals is decidable.
Using first order theory on reals

- **Decision problem**: does player 1 has a strategy which guarantees payoff $> 0$?
- **Theorem [Shapley 53]**: it is enough to consider stationary (but not deterministic) strategies $\sigma : S \rightarrow D (I)$.
- **First order theory on reals**: well-formed formula with rational constants $a/b$, arithmetic operations $*$ and $+$, variables $x_1, x_2, \ldots, x_n$, comparison $\leq$, quantifiers $\exists$ and $\forall$, boolean operators $\neg, \land, \lor$ and parentheses $( and )$. $\exists x, 3x^6 - 4x^2 + 3 = 0$.
- **Theorem [Tarski, 51]**: quantifier elimination. Truth of first order formula on reals is decidable.
- **Corollary [Chatterjee, 06]**: whether player 1 can guarantee payoff $> 0$ is decidable. Exponential time, polynomial space.
Using first order theory on reals

- Reduction of the decision problem to FO on reals.
Using first order theory on reals

- **Reduction of the decision problem** to FO on reals.

- **Shapley**: values $\text{val} : S \rightarrow \mathbb{R}$ are the unique fixpoint of a contracting operator $\mathbb{R}^S \rightarrow \mathbb{R}^S$. 
Using first order theory on reals

- **Reduction of the decision problem** to FO on reals.
- **Shapley**: values $\text{val} : S \rightarrow \mathbb{R}$ are the unique fixpoint of a contracting operator $\mathbb{R}^S \rightarrow \mathbb{R}^S$.
- **Fixed stationary strategies** $\sigma : S \rightarrow \mathcal{D}(I)$ and $\tau : S \rightarrow \mathcal{D}(J)$, expected payoff $\text{val}(\sigma, \tau) : S \rightarrow \mathbb{R}$ is the unique solution to $\text{val}(s) = r(s) + \lambda \sum_{i,j,u} \sigma(i)\tau(j)p(s, i, j, u)\text{val}(u)$ (**)
Using first order theory on reals

- Reduction of the decision problem to FO on reals.
- **Shapley**: values \( \text{val} : S \to \mathbb{R} \) are the unique fixpoint of a contracting operator \( \mathbb{R}^S \to \mathbb{R}^S \).
- Fixed stationary strategies \( \sigma : S \to \mathcal{D}(I) \) and \( \tau : S \to \mathcal{D}(J) \), expected payoff \( \text{val}(\sigma, \tau) : S \to \mathbb{R} \) is the unique solution to
  \[
  \text{val}(s) = r(s) + \lambda \sum_{i,j,u} \sigma(i)\tau(j)p(s,i,j,u)\text{val}(u). \quad (**) 
  \]
- \( \exists \sigma : S \to \mathcal{D}(I), \forall \tau : S \to \mathcal{D}(J), \exists \nu : S \to [0,1], (\forall s \in S, (**)) \land (\nu(s_0) > 0) \).
Algorithms and Games

Solving Simple Stochastic Games

Solving Stochastic Games

Solving Stochastic Games with Signals

Conclusion
Stochastic Games with Signals

- Players do not observe the current state of the game, they receive signals. Players only observe their signals.
Stochastic Games with Signals

- Players do not observe the current state of the game, they receive signals. Players only observe their signals.

- Example. Actions \( \{0, 1, g_1, g_2\} \) for 1 and \( \{0, 1\} \) for 2. Signals \( \{\alpha, \beta\} \) for 1 and \( \{\cdot\} \) for 2.
Players do not observe the current state of the game, they receive signals. Players only observe their signals.

Example. Actions \{0, 1, g_1, g_2\} for 1 and \{0, 1\} for 2. Signals \{\alpha, \beta\} for 1 and \{\cdot\} for 2.
Strategies

- Actions $I$ and $J$ for player 1 and 2.
Strategies

- **Actions** $I$ and $J$ for player 1 and 2.
- **Signals** $C$ and $D$ for player 1 and 2. Contain actions.
Strategies

- **Actions** \( I \) and \( J \) for player 1 and 2.
- **Signals** \( C \) and \( D \) for player 1 and 2. Contain actions.
- **Behavioral strategy** for player 1 \( \sigma : C^* \rightarrow D(I) \).
Strategies

- **Actions** $I$ and $J$ for player 1 and 2.
- **Signals** $C$ and $D$ for player 1 and 2. Contain actions.
- **Behavioral strategy** for player 1 $\sigma : C^* \rightarrow D(I)$.
- **Decision problem**: does player 1 has a strategy $\sigma$ for reaching $t$ with probability more than $\frac{1}{2}$?
The lonely blind player

- Easy case: the lonely blind player.
The lonely blind player

- Easy case: the lonely blind player.
- One player, blind and alone. For player 1 signals = actions = \( I \). For player 2 actions = singleton.
The lonely blind player

- Easy case: the lonely blind player.
- One player, blind and alone. For player 1 signals = actions = $I$. For player 2 actions = singleton.
- Deterministic strategy: infinite sequence of actions in $I^\mathbb{N}$.
The lonely blind player

- Easy case: the lonely blind player.
- One player, blind and alone. For player 1 signals = actions = \( I \). For player 2 actions = singleton.
- **Deterministic strategy**: infinite sequence of actions in \( I^\mathbb{N} \).
- **Value** \( \sup_{u \in I^\mathbb{N}} \mathbb{P}^u_s(\exists n \in \mathbb{N}, K_n \in T) \). Same value if \( u \) finite.
The lonely blind player

- Easy case: the lonely blind player.
- One player, blind and alone. For player 1 signals = actions = \( I \). For player 2 actions = singleton.
- Deterministic strategy: infinite sequence of actions in \( I^\mathbb{N} \).
- Value \( \sup_{u \in I^\mathbb{N}} \mathbb{P}_s(\exists n \in \mathbb{N}, K_n \in T) \). Same value if \( u \) finite.
- Decision problem: does player 1 has a strategy for reaching \( t \) with probability \( > \frac{1}{2} \).
The lonely blind player

- **Easy case:** the lonely blind player.
- One player, blind and alone. For player 1 signals = actions = \( I \). For player 2 actions = singleton.
- **Deterministic strategy:** infinite sequence of actions in \( I^\mathbb{N} \).
- **Value** \( \sup_{u \in I^\mathbb{N}} \mathbb{P}_s(\exists n \in \mathbb{N}, K_n \in T) \). Same value if \( u \) finite.
- **Decision problem:** does player 1 has a strategy for reaching \( t \) with probability \( > \frac{1}{2} \).
- Enumeration of finite words \( u \)?
Bad news

▶ Theorem: it is undecidable whether player 1 can win with probability > 1/2 in a lonely blind game.
Bad news

- Theorem: it is undecidable whether player 1 can win with probability $> 1/2$ in a lonely blind game.
- Unlimited memory, unlimited speed.
Bad news

- Theorem: it is undecidable whether player 1 can win with probability $> 1/2$ in a lonely blind game.
- Unlimited memory, unlimited speed.
- Proof: reduction to Post correspondence problem. Actions in the game = indices of the PCP instance. Reverse binary encoding, strategy wins with probability
\[
\frac{1}{2} u_1 u_2 \cdots u_n + (1 - \frac{1}{2}) v_1 v_2 \cdots v_n.
\] Strategies win with proba
\[
\frac{1}{2} \text{ iff } u_1 u_2 \cdots u_n = v_1 v_2 \cdots v_n.
\]
Decidable questions for stochastic games with signals

▶ Another Decision problem: does player 1 has a strategy for winning with probability exactly 1?
Decidable questions for stochastic games with signals

- **Another Decision problem**: does player 1 have a strategy for winning with probability exactly 1?

- **Theorem [Bertrand, Genest, G.]**: if not, player 2 has a strategy with finite memory, whose size is doubly-exponential in the number of states.
Decidable questions for stochastic games with signals

- **Another Decision problem**: does player 1 has a strategy for winning with probability exactly 1?
- **Theorem [Bertrand, Genest, G.]**: if not, player 2 has a strategy with finite memory, whose size is doubly-exponential in the number of states.
- **Corollary**: this decision problem is decidable in doubly-exponential time.
Another Decision problem: does player 1 has a strategy for winning with probability exactly 1?

Theorem [Bertrand, Genest, G.]: if not, player 2 has a strategy with finite memory, whose size is doubly-exponential in the number of states.

Corollary: this decision problem is decidable in doubly-exponential time.

Remark: the same decision problem is undecidable for stochastic games with Büchi conditions.
Conclusion

- Stochastic games on **infinite graphs** finitely presented: tree-automata techniques, game reductions, FO.
Conclusion

- Stochastic games on infinite graphs finitely presented: tree-automata techniques, game reductions, FO.
- Algorithm rely on precise description (quantity of memory, finite description) of optimal strategies.
Conclusion

- Stochastic games on \textit{infinite graphs} finitely presented: tree-automata techniques, game reductions, FO.
- \textbf{Algorithm rely on precise description} (quantity of memory, finite description) of \textit{optimal strategies}.
- Finding \textit{subclasses of stochastic games with signals} with decidable decision problems.
Conclusion

- Stochastic games on **infinite graphs** finitely presented: tree-automata techniques, game reductions, FO.
- **Algorithm rely on precise description** (quantity of memory, finite description) of **optimal strategies**.
- Finding **subclasses of stochastic games with signals** with decidable decision problems.
- Avoid reduction to first order logic.
Conclusion

- Stochastic games on infinite graphs finitely presented: tree-automata techniques, game reductions, FO.
- Algorithm rely on precise description (quantity of memory, finite description) of optimal strategies.
- Finding subclasses of stochastic games with signals with decidable decision problems.
- Avoid reduction to first order logic.
- Finding a polynomial-time algorithm for simple stochastic games.