Linear Equations with Min and Max Operators

Krish Chatterjee[1], **Ruichen Luo**[1], Raimundo Saona[1], Jakub Svoboda[1]

[1]Institute of Science and Technology Austria (ISTA)
Klosterneuburg, Austria

Nov 4, 2025, Kaiserslautern, Germany

Outline

1. Introduction
   - Stochastic games
   - Optimization problem
2. Motivating examples and restrictive conditions
   - Motivating examples
   - Restrictive conditions
3. Complexity
   - Complexity of the decision problem
   - Complexity of checking the conditions
4. Algorithms
   - Classic algorithms for halting SSGs
   - Algorithms for absolutely halting subclass
   - Algorithms for halting subclass

Games on graph

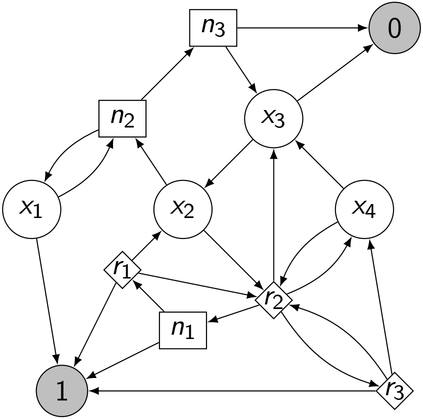

Figure 1: A graph game with $2^{1/2}$ players: $\square$–Min, $\bigcirc$–Max, and $\Diamond$–Random.

Stochastic games: Preliminaries

- $2^{1/2}$-player (Min, Max, Random)
- Application: Modelling controller synthesis
  - Player Min: Controller
  - Player Max: Dynamics
  - Player Random: Probabilistic system, fairness, $\cdots$
- More applications:
  - Decision-making under uncertainty
  - Robust stochastic control or optimization
  - $\cdots$

Stochastic games: Preliminaries

- $2^{1/2}$-player (MIN, MAX, RANDOM)
- Application: Modelling controller synthesis
  - Player MIN: Controller
  - Player MAX: Dynamics
  - Player RANDOM: Probabilistic system, fairness, $\cdots$
- More applications:
  - Decision-making under uncertainty
  - Robust stochastic control or optimization
  - $\cdots$

Stochastic games: Preliminaries

- $2^{1/2}$-player (Min, Max, Random)
- Application: Modelling controller synthesis
  - Player Min: Controller
  - Player Max: Dynamics
  - Player Random: Probabilistic system, fairness, $\cdots$
- More applications:
  - Decision-making under uncertainty
  - Robust stochastic control or optimization
  - $\cdots$

Stochastic games: Preliminaries

- $2^{1/2}$-player (Min, Max, Random)
- Application: Modelling controller synthesis
  - Player Min: Controller
  - Player Max: Dynamics
  - Player Random: Probabilistic system, fairness, $\cdots$
- More applications:
  - Decision-making under uncertainty
  - Robust stochastic control or optimization
  - $\cdots$

Stochastic games: Preliminaries

- $2^{1/2}$-player (Min, Max, Random)
- Application: Modelling controller synthesis
  - Player Min: Controller
  - Player Max: Dynamics
  - Player Random: Probabilistic system, fairness, $\cdots$
- More applications:
  - Decision-making under uncertainty
  - Robust stochastic control or optimization
  - $\cdots$

Stochastic games: Preliminaries

- $2^{1/2}$-player ($\textsc{Min}$, $\textsc{Max}$, $\textsc{Random}$)
- Application: Modelling controller synthesis
  - Player $\textsc{Min}$: Controller
  - Player $\textsc{Max}$: Dynamics
  - Player $\textsc{Random}$: Probabilistic system, fairness, $\cdots$
- More applications:
  - Decision-making under uncertainty
  - Robust stochastic control or optimization
  - $\cdots$

Stochastic games: Preliminaries

- $2^{1/2}$-player (Min, Max, Random)
- Application: Modelling controller synthesis
  - Player Min: Controller
  - Player Max: Dynamics
  - Player Random: Probabilistic system, fairness, $\cdots$
- More applications:
  - Decision-making under uncertainty
  - Robust stochastic control or optimization
  - $\cdots$

Stochastic games: Preliminaries

- $2^{1/2}$-player ($\textsc{Min}$, $\textsc{Max}$, $\textsc{Random}$)
- Application: Modelling controller synthesis
  - Player $\textsc{Min}$: Controller
  - Player $\textsc{Max}$: Dynamics
  - Player $\textsc{Random}$: Probabilistic system, fairness, $\cdots$
- More applications:
  - Decision-making under uncertainty
  - Robust stochastic control or optimization
  - $\cdots$

Stochastic games: Preliminaries

- $2^{1/2}$-player (Min, Max, Random)
- Application: Modelling controller synthesis
  - Player Min: Controller
  - Player Max: Dynamics
  - Player Random: Probabilistic system, fairness, $\cdots$
- More applications:
  - Decision-making under uncertainty
  - Robust stochastic control or optimization
  - $\cdots$

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node *i*?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Anne Condon. "The complexity of stochastic games". In: *Information and Computation* 96.2 (1992), pp. 203–224.

[2] Ibid.

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node i?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Condon, "The complexity of stochastic games".
[2] Ibid.

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node i?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Condon, "The complexity of stochastic games".
[2] Ibid.

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node i?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Condon, "The complexity of stochastic games".
[2] Ibid.

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node i?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Condon, "The complexity of stochastic games".
[2] Ibid.

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node i?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Condon, "The complexity of stochastic games".
[2] Ibid.

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node i?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Condon, "The complexity of stochastic games".
[2] Ibid.

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node *i*?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Condon, "The complexity of stochastic games".
[2] Ibid.

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node *i*?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Condon, "The complexity of stochastic games".

[2] Ibid.

Simple stochastic games (SSGs)

- Restrictions:
  - Turn-based
  - Two-player zero-sum
  - Reachability
  - Perfect information
  - (Halting)
- There exists a Nash equilibrium in pure memory-less strategy.[1]
- Decision problem:
  Is the winning probability (of $\textsc{Max}$) $< \beta$ starting from node *i*?
  Complexity: Between UP $\cap$ coUP and PTIME.
- Fixpoint characterization of winning probability:
  A linear system with min/max operators.[2]

[1] Condon, "The complexity of stochastic games".
[2] Ibid.

Outline

1. Introduction
   - Stochastic games
   - Optimization problem
2. Motivating examples and restrictive conditions
   - Motivating examples
   - Restrictive conditions
3. Complexity
   - Complexity of the decision problem
   - Complexity of checking the conditions
4. Algorithms
   - Classic algorithms for halting SSGs
   - Algorithms for absolutely halting subclass
   - Algorithms for halting subclass

Linear equations with min and max operators

We consider linear equations with min/max operators of $\mathbf{x} \in \mathbb{R}^n$:

$$\begin{cases} x_i = \min\limits_{l \in \mathcal{N}(i)} x_l, \quad 1 \le i \le n_1, & \text{(1a)} \\ x_j = \max\limits_{l \in \mathcal{N}(j)} x_l, \quad n_1 < j \le n_1 + n_2, & \text{(1b)} \\ x_k = \mathbf{q}_k^\mathsf{T} \mathbf{x} + b_k, \quad n_1 + n_2 < k \le n, & \text{(1c)} \end{cases}$$

where $(n_1, n_2, n, \mathcal{N}, \mathbf{q}, \mathbf{b})$ satisfies the following conditions:
(a) $n_1, n_2, n \in \mathbb{N}$, $n \ge n_1 + n_2$,
(b) $\varnothing \subsetneq \mathcal{N}(i) \subseteq [n]$ for $i \in [n_1 + n_2]$,
(c) $\mathbf{q}_k \in \mathbb{R}^n$ for $n_1 + n_2 < k \le n$, and
(d) $\mathbf{b} = [0, \ldots, 0, b_{n_1+n_2+1}, \ldots, b_n]^\mathsf{T} \in \mathbb{R}^n$.

Definition 1 (Decision problem)

Given Equation (1), an index $i \in [n]$, and a threshold $\beta \in \mathbb{R}$, determine whether there exists a solution $\mathbf{x}$ such that $x_i < \beta$.

Linear equations with min and max operators

We consider linear equations with min/max operators of $\mathbf{x} \in \mathbb{R}^n$:

$$\begin{cases} x_i = \min\limits_{l \in \mathcal{N}(i)} x_l, \quad 1 \le i \le n_1, & \text{(1a)} \\ x_j = \max\limits_{l \in \mathcal{N}(j)} x_l, \quad n_1 < j \le n_1 + n_2, & \text{(1b)} \\ x_k = \mathbf{q}_k^{\mathsf{T}} \mathbf{x} + b_k, \quad n_1 + n_2 < k \le n, & \text{(1c)} \end{cases}$$

where $(n_1, n_2, n, \mathcal{N}, \mathbf{q}, \mathbf{b})$ satisfies the following conditions:

(a) $n_1, n_2, n \in \mathbb{N}$, $n \ge n_1 + n_2$,
(b) $\varnothing \subsetneq \mathcal{N}(i) \subseteq [n]$ for $i \in [n_1 + n_2]$,
(c) $\mathbf{q}_k \in \mathbb{R}^n$ for $n_1 + n_2 < k \le n$, and
(d) $\mathbf{b} = [0, \ldots, 0, b_{n_1+n_2+1}, \ldots, b_n]^{\mathsf{T}} \in \mathbb{R}^n$.

Definition 1 (Decision problem)

Given Equation (1), an index $i \in [n]$, and a threshold $\beta \in \mathbb{R}$, determine whether there exists a solution $\mathbf{x}$ such that $x_i < \beta$.

Linear equations with min and max operators

We consider linear equations with min/max operators of $\mathbf{x} \in \mathbb{R}^n$:

$$\begin{cases} x_i = \min\limits_{l \in \mathcal{N}(i)} x_l, \quad 1 \le i \le n_1, & \text{(1a)} \\ x_j = \max\limits_{l \in \mathcal{N}(j)} x_l, \quad n_1 < j \le n_1 + n_2, & \text{(1b)} \\ x_k = \mathbf{q}_k^\mathsf{T} \mathbf{x} + b_k, \quad n_1 + n_2 < k \le n, & \text{(1c)} \end{cases}$$

where $(n_1, n_2, n, \mathcal{N}, \mathbf{q}, \mathbf{b})$ satisfies the following conditions:

(a) $n_1, n_2, n \in \mathbb{N}$, $n \ge n_1 + n_2$,
(b) $\varnothing \subsetneq \mathcal{N}(i) \subseteq [n]$ for $i \in [n_1 + n_2]$,
(c) $\mathbf{q}_k \in \mathbb{R}^n$ for $n_1 + n_2 < k \le n$, and
(d) $\mathbf{b} = [0, \ldots, 0, b_{n_1+n_2+1}, \ldots, b_n]^\mathsf{T} \in \mathbb{R}^n$.

Definition 1 (Decision problem)

Given Equation (1), an index $i \in [n]$, and a threshold $\beta \in \mathbb{R}$, determine whether there exists a solution $\mathbf{x}$ such that $x_i < \beta$.

Outline

1. Introduction
   - Stochastic games
   - Optimization problem
2. Motivating examples and restrictive conditions
   - Motivating examples
   - Restrictive conditions
3. Complexity
   - Complexity of the decision problem
   - Complexity of checking the conditions
4. Algorithms
   - Classic algorithms for halting SSGs
   - Algorithms for absolutely halting subclass
   - Algorithms for halting subclass

Verifying neural networks[3]

- Verifying multi-layer perceptron (MLP):
  - Given an n-layer MLP with scalar output, decide whether

$$x^{\text{out}} < \beta, \text{ for all } \mathbf{x}^{\text{in}} \in [0,1]^d.$$

  - Activation function σ: ReLU or Maxout.
-

$$\begin{cases} \mathbf{x}^{\text{in}} = \max\{0, \min\{\mathbf{x}^{\text{in}}, 1\}\}\,, \\ \mathbf{x}^{\text{Layer }1} = \sigma(\mathbf{Q}^{\text{Layer }1}\mathbf{x}^{\text{in}} + \mathbf{b}^{\text{Layer }1})\,, \\ \vdots \\ \mathbf{x}^{\text{Layer }n} = \sigma(\mathbf{Q}^{\text{Layer }n}\mathbf{x}^{\text{Layer }n-1} + \mathbf{b}^{\text{Layer }n})\,, \\ x^{\text{out}} = {\mathbf{q}^{\text{out}}}^{\mathsf{T}}\mathbf{x}^{\text{Layer }n} + b^{\text{out}}. \end{cases}$$

[3] Guy Katz et al. "Reluplex: An efficient SMT solver for verifying deep neural networks". In: *International conference on computer aided verification*. Springer. 2017, pp. 97–117.

Verifying neural networks[3]

- Verifying multi-layer perceptron (MLP):
  - Given an n-layer MLP with scalar output, decide whether

$$x^{\text{out}} < \beta, \text{ for all } \mathbf{x}^{\text{in}} \in [0,1]^d.$$

  - Activation function σ: ReLU or Maxout.
-

$$\begin{cases} \mathbf{x}^{\text{in}} = \max\{0, \min\{\mathbf{x}^{\text{in}}, 1\}\}, \\ \mathbf{x}^{\text{Layer } 1} = \sigma(\mathbf{Q}^{\text{Layer } 1}\mathbf{x}^{\text{in}} + \mathbf{b}^{\text{Layer } 1}), \\ \vdots \\ \mathbf{x}^{\text{Layer } n} = \sigma(\mathbf{Q}^{\text{Layer } n}\mathbf{x}^{\text{Layer } n-1} + \mathbf{b}^{\text{Layer } n}), \\ x^{\text{out}} = \mathbf{q}^{\text{out}^{\mathsf{T}}}\mathbf{x}^{\text{Layer } n} + b^{\text{out}}. \end{cases}$$

[3] Katz et al., "Reluplex: An efficient SMT solver for verifying deep neural networks".

Verifying neural networks[3]

- Verifying multi-layer perceptron (MLP):
  - Given an n-layer MLP with scalar output, decide whether

$$x^{\text{out}} < \beta, \text{ for all } \mathbf{x}^{\text{in}} \in [0,1]^d.$$

  - Activation function σ: ReLU or Maxout.
-

$$\begin{cases} \mathbf{x}^{\text{in}} = \max\{0, \min\{\mathbf{x}^{\text{in}}, 1\}\}, \\ \mathbf{x}^{\text{Layer } 1} = \sigma(\mathbf{Q}^{\text{Layer } 1}\mathbf{x}^{\text{in}} + \mathbf{b}^{\text{Layer } 1}), \\ \vdots \\ \mathbf{x}^{\text{Layer } n} = \sigma(\mathbf{Q}^{\text{Layer } n}\mathbf{x}^{\text{Layer } n-1} + \mathbf{b}^{\text{Layer } n}), \\ x^{\text{out}} = \mathbf{q}^{\text{out}^{\mathsf{T}}}\mathbf{x}^{\text{Layer } n} + b^{\text{out}}. \end{cases}$$

[3] Katz et al., "Reluplex: An efficient SMT solver for verifying deep neural networks".

Verifying neural networks[3]

- Verifying multi-layer perceptron (MLP):
  - Given an n-layer MLP with scalar output, decide whether

$$x^{\text{out}} < \beta, \text{ for all } \mathbf{x}^{\text{in}} \in [0,1]^d.$$

  - Activation function σ: ReLU or Maxout.
-

$$\begin{cases} \mathbf{x}^{\text{in}} = \max\{0, \min\{\mathbf{x}^{\text{in}}, 1\}\}, \\ \mathbf{x}^{\text{Layer } 1} = \sigma(\mathbf{Q}^{\text{Layer } 1}\mathbf{x}^{\text{in}} + \mathbf{b}^{\text{Layer } 1}), \\ \vdots \\ \mathbf{x}^{\text{Layer } n} = \sigma(\mathbf{Q}^{\text{Layer } n}\mathbf{x}^{\text{Layer } n-1} + \mathbf{b}^{\text{Layer } n}), \\ x^{\text{out}} = {\mathbf{q}^{\text{out}}}^{\mathsf{T}} \mathbf{x}^{\text{Layer } n} + b^{\text{out}}. \end{cases}$$

[3] Katz et al., "Reluplex: An efficient SMT solver for verifying deep neural networks".

Verifying neural networks[3]

- Verifying multi-layer perceptron (MLP):
  - Given an n-layer MLP with scalar output, decide whether

$$x^{\text{out}} < \beta, \text{ for all } \mathbf{x}^{\text{in}} \in [0,1]^d.$$

  - Activation function σ: ReLU or Maxout.
- $$\begin{cases} \mathbf{x}^{\text{in}} = \max\{0, \min\{\mathbf{x}^{\text{in}}, 1\}\}, \\ \mathbf{x}^{\text{Layer } 1} = \sigma(\mathbf{Q}^{\text{Layer } 1}\mathbf{x}^{\text{in}} + \mathbf{b}^{\text{Layer } 1}), \\ \vdots \\ \mathbf{x}^{\text{Layer } n} = \sigma(\mathbf{Q}^{\text{Layer } n}\mathbf{x}^{\text{Layer } n-1} + \mathbf{b}^{\text{Layer } n}), \\ x^{\text{out}} = {\mathbf{q}^{\text{out}}}^{\mathsf{T}} \mathbf{x}^{\text{Layer } n} + b^{\text{out}}. \end{cases}$$

[3] Katz et al., "Reluplex: An efficient SMT solver for verifying deep neural networks".

Capital preservation[4]

- Games played in financial market cycles:
  - Controller – to decide the investment of the assets;
  - Market – to impact the asset prices.
-

$$\begin{cases} \mathbf{x}_{1,1} = \max\{\mathbf{Q}_{1,i}^{\top}\mathbf{x}_{T,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{1,2} = \min\{\mathbf{Q}_{2,j}^{\top}\mathbf{x}_{1,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,, \\ \qquad\vdots \\ \mathbf{x}_{T,1} = \max\{\mathbf{Q}_{1,i}^{\top}\mathbf{x}_{T-1,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{T,2} = \min\{\mathbf{Q}_{2,j}^{\top}\mathbf{x}_{T,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,. \end{cases}$$

[4] Harry Markowitz. "Portfolio Selection". In: *The Journal of Finance* 7.1 (1952), pp. 77–91. URL: http://www.jstor.org/stable/2975974.

Capital preservation[4]

- Games played in financial market cycles:
  - Controller – to decide the investment of the assets;
  - Market – to impact the asset prices.
-

$$
\begin{cases}
\mathbf{x}_{1,1} = \max\{\mathbf{Q}_{1,i}^\top \mathbf{x}_{T,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\
\mathbf{x}_{1,2} = \min\{\mathbf{Q}_{2,j}^\top \mathbf{x}_{1,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,, \\
\quad\vdots \\
\mathbf{x}_{T,1} = \max\{\mathbf{Q}_{1,i}^\top \mathbf{x}_{T-1,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\
\mathbf{x}_{T,2} = \min\{\mathbf{Q}_{2,j}^\top \mathbf{x}_{T,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,.
\end{cases}
$$

[4] Markowitz, “Portfolio Selection”.

Capital preservation[4]

- Games played in financial market cycles:
  - Controller – to decide the investment of the assets;
  - Market – to impact the asset prices.
-

$$\begin{cases} \mathbf{x}_{1,1} = \max\{\mathbf{Q}_{1,i}^\top \mathbf{x}_{T,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{1,2} = \min\{\mathbf{Q}_{2,j}^\top \mathbf{x}_{1,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,, \\ \quad \vdots \\ \mathbf{x}_{T,1} = \max\{\mathbf{Q}_{1,i}^\top \mathbf{x}_{T-1,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{T,2} = \min\{\mathbf{Q}_{2,j}^\top \mathbf{x}_{T,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,. \end{cases}$$

[4] Markowitz, “Portfolio Selection”.

Capital preservation[4]

- Games played in financial market cycles:
  - Controller – to decide the investment of the assets;
  - Market – to impact the asset prices.
- $$\begin{cases} \mathbf{x}_{1,1} = \max\{\mathbf{Q}_{1,i}^{\top}\mathbf{x}_{T,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{1,2} = \min\{\mathbf{Q}_{2,j}^{\top}\mathbf{x}_{1,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,, \\ \quad\vdots \\ \mathbf{x}_{T,1} = \max\{\mathbf{Q}_{1,i}^{\top}\mathbf{x}_{T-1,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{T,2} = \min\{\mathbf{Q}_{2,j}^{\top}\mathbf{x}_{T,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,. \end{cases}$$

[4] Markowitz, "Portfolio Selection".

Capital preservation[4]

- Games played in financial market cycles:
  - Controller – to decide the investment of the assets;
  - Market – to impact the asset prices.
- $$\begin{cases} \mathbf{x}_{1,1} = \max\{\mathbf{Q}_{1,i}^{\top}\mathbf{x}_{T,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{1,2} = \min\{\mathbf{Q}_{2,j}^{\top}\mathbf{x}_{1,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,, \\ \quad\vdots \\ \mathbf{x}_{T,1} = \max\{\mathbf{Q}_{1,i}^{\top}\mathbf{x}_{T-1,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{T,2} = \min\{\mathbf{Q}_{2,j}^{\top}\mathbf{x}_{T,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,. \end{cases}$$

[4] Markowitz, "Portfolio Selection".

Capital preservation[4]

- Games played in financial market cycles:
  - Controller – to decide the investment of the assets;
  - Market – to impact the asset prices.
- $$\begin{cases} \mathbf{x}_{1,1} = \max\{\mathbf{Q}_{1,i}^{\top}\,\mathbf{x}_{T,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{1,2} = \min\{\mathbf{Q}_{2,j}^{\top}\,\mathbf{x}_{1,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,, \\ \qquad\vdots \\ \mathbf{x}_{T,1} = \max\{\mathbf{Q}_{1,i}^{\top}\,\mathbf{x}_{T-1,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{T,2} = \min\{\mathbf{Q}_{2,j}^{\top}\,\mathbf{x}_{T,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,. \end{cases}$$

[4]Markowitz, “Portfolio Selection”.

Capital preservation[4]

- Games played in financial market cycles:
  - Controller – to decide the investment of the assets;
  - Market – to impact the asset prices.
- $$\begin{cases} \mathbf{x}_{1,1} = \max\{\mathbf{Q}_{1,i}^{\top}\mathbf{x}_{T,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{1,2} = \min\{\mathbf{Q}_{2,j}^{\top}\mathbf{x}_{1,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,, \\ \quad\vdots \\ \mathbf{x}_{T,1} = \max\{\mathbf{Q}_{1,i}^{\top}\mathbf{x}_{T-1,2} + \mathbf{b}_{1,i} \mid i \in [m]\}\,, \\ \mathbf{x}_{T,2} = \min\{\mathbf{Q}_{2,j}^{\top}\mathbf{x}_{T,1} + \mathbf{b}_{2,j} \mid j \in [\ell]\}\,. \end{cases}$$

[4] Markowitz, "Portfolio Selection".

Evolution in ecosystem

- In an ecosystem with k species, we monitor their populations:
  - Current population – $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$.
  - Next population – $\mathbf{x}' = \max\{\mathbf{Q}\mathbf{x} + \mathbf{b}, 0\}$,
    where
    - $\mathbf{Q} \in \mathbb{R}^{k \times k}$ – internal interactions,
    - $\mathbf{b} \in \mathbb{R}^k$ – external interventions.
- Ecological balance:

$$\mathbf{x} = \max\{\mathbf{Q}\mathbf{x} + \mathbf{b}, 0\}.$$

Evolution in ecosystem

- In an ecosystem with k species, we monitor their populations:
  - Current population – $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$.
  - Next population – $\mathbf{x}' = \max\{\mathbf{Qx} + \mathbf{b}, 0\}$, where
    - $\mathbf{Q} \in \mathbb{R}^{k \times k}$ – internal interactions,
    - $\mathbf{b} \in \mathbb{R}^k$ – external interventions.
- Ecological balance:

$$\mathbf{x} = \max\{\mathbf{Qx} + \mathbf{b}, 0\}.$$

Evolution in ecosystem

- In an ecosystem with k species, we monitor their populations:
  - Current population – $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$.
  - Next population – $\mathbf{x}' = \max\{\mathbf{Q}\mathbf{x} + \mathbf{b}, 0\}$,
    where
    - $\mathbf{Q} \in \mathbb{R}^{k \times k}$ – internal interactions,
    - $\mathbf{b} \in \mathbb{R}^k$ – external interventions.
- Ecological balance:

$$\mathbf{x} = \max\{\mathbf{Q}\mathbf{x} + \mathbf{b}, 0\}.$$

Evolution in ecosystem

- In an ecosystem with k species, we monitor their populations:
  - Current population – $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$.
  - Next population – $\mathbf{x}' = \max\{\mathbf{Q}\mathbf{x} + \mathbf{b}, 0\}$, where
    - $\mathbf{Q} \in \mathbb{R}^{k \times k}$ – internal interactions,
    - $\mathbf{b} \in \mathbb{R}^k$ – external interventions.
- Ecological balance:

$$\mathbf{x} = \max\{\mathbf{Q}\mathbf{x} + \mathbf{b}, 0\}.$$

Evolution in ecosystem

- In an ecosystem with k species, we monitor their populations:
  - Current population – $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$.
  - Next population – $\mathbf{x}' = \max\{\mathbf{Q}\mathbf{x} + \mathbf{b}, 0\}$,
    where
    - $\mathbf{Q} \in \mathbb{R}^{k \times k}$ – internal interactions,
    - $\mathbf{b} \in \mathbb{R}^k$ – external interventions.
- Ecological balance:

$$\mathbf{x} = \max\{\mathbf{Q}\mathbf{x} + \mathbf{b}, 0\}.$$

Evolution in ecosystem

- In an ecosystem with k species, we monitor their populations:
  - Current population – $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$.
  - Next population – $\mathbf{x}' = \max\{\mathbf{Qx} + \mathbf{b}, 0\}$,
    where
    - $\mathbf{Q} \in \mathbb{R}^{k \times k}$ – internal interactions,
    - $\mathbf{b} \in \mathbb{R}^k$ – external interventions.
- Ecological balance:

$$\mathbf{x} = \max\{\mathbf{Qx} + \mathbf{b}, 0\}.$$

Outline

1. Introduction
   - Stochastic games
   - Optimization problem
2. Motivating examples and restrictive conditions
   - Motivating examples
   - Restrictive conditions
3. Complexity
   - Complexity of the decision problem
   - Complexity of checking the conditions
4. Algorithms
   - Classic algorithms for halting SSGs
   - Algorithms for absolutely halting subclass
   - Algorithms for halting subclass

Notations

Recall

$$
\begin{cases}
x_i = \min\limits_{l \in \mathcal{N}(i)} x_l, & 1 \leq i \leq n_1, \\
x_j = \max\limits_{l \in \mathcal{N}(j)} x_l, & n_1 < j \leq n_1 + n_2, \\
x_k = \mathbf{q}_k^\mathsf{T} \mathbf{x} + b_k, & n_1 + n_2 < k \leq n.
\end{cases}
$$

- $\delta_{i,j}$ is 1 if $i = j$ and 0 otherwise.
- $\mathbf{e}_i := [\delta_{i,1}, \ldots, \delta_{i,n}]^\mathsf{T},\ i \in [n]$.
- $\mathcal{Q} := \left\{ [\mathbf{e}_{\ell_1}, \ldots, \mathbf{e}_{\ell_{n_1+n_2}}, \mathbf{q}_{n_1+n_2+1}, \ldots, \mathbf{q}_n]^\mathsf{T} \,\middle|\, \ell_j \in \mathcal{N}(j) \text{ for } j \in [n_1 + n_2] \right\}$.
- $\mathbf{conv}(\mathcal{Q}) := \left\{ \sum_{i \in \mathcal{I}} \alpha_i \mathbf{Q}_i \,\middle|\, \sum_{i \in \mathcal{I}} \alpha_i = 1 \text{ and } \forall\, i \in \mathcal{I},\ \mathbf{Q}_i \in \mathcal{Q} \wedge \alpha_i \geq 0 \right\}$.

Notations

Recall

$$\begin{cases} x_i = \min_{l \in \mathcal{N}(i)} x_l, & 1 \leq i \leq n_1, \\ x_j = \max_{l \in \mathcal{N}(j)} x_l, & n_1 < j \leq n_1 + n_2, \\ x_k = \mathbf{q}_k^{\mathsf{T}} \mathbf{x} + b_k, & n_1 + n_2 < k \leq n. \end{cases}$$

- $\delta_{i,j}$ is 1 if $i = j$ and 0 otherwise.
- $\mathbf{e}_i \coloneqq [\delta_{i,1}, \ldots, \delta_{i,n}]^{\mathsf{T}},\ i \in [n]$.
- $\mathcal{Q} \coloneqq \left\{ [\mathbf{e}_{\ell_1}, \ldots, \mathbf{e}_{\ell_{n_1+n_2}}, \mathbf{q}_{n_1+n_2+1}, \ldots, \mathbf{q}_n]^{\mathsf{T}} \,\middle|\, \ell_j \in \mathcal{N}(j) \text{ for } j \in [n_1+n_2] \right\}$.
- $\mathbf{conv}(\mathcal{Q}) \coloneqq \left\{ \sum_{i \in \mathcal{I}} \alpha_i \mathbf{Q}_i \,\middle|\, \sum_{i \in \mathcal{I}} \alpha_i = 1 \text{ and } \forall\ i \in \mathcal{I},\ \mathbf{Q}_i \in \mathcal{Q} \wedge \alpha_i \geq 0 \right\}$.

Notations

Recall

$$\begin{cases} x_i = \min\limits_{l \in \mathcal{N}(i)} x_l, & 1 \leq i \leq n_1, \\ x_j = \max\limits_{l \in \mathcal{N}(j)} x_l, & n_1 < j \leq n_1 + n_2, \\ x_k = \mathbf{q}_k^\mathsf{T} \mathbf{x} + b_k, & n_1 + n_2 < k \leq n. \end{cases}$$

- $\delta_{i,j}$ is 1 if $i = j$ and 0 otherwise.
- $\mathbf{e}_i \coloneqq [\delta_{i,1}, \ldots, \delta_{i,n}]^\mathsf{T},\ i \in [n]$.
- $\mathcal{Q} \coloneqq \left\{ [\mathbf{e}_{\ell_1}, \ldots, \mathbf{e}_{\ell_{n_1+n_2}}, \mathbf{q}_{n_1+n_2+1}, \ldots, \mathbf{q}_n]^\mathsf{T} \,\middle|\, \ell_j \in \mathcal{N}(j) \text{ for } j \in [n_1 + n_2] \right\}$.
- $\mathbf{conv}(\mathcal{Q}) \coloneqq \left\{ \sum_{i \in \mathcal{I}} \alpha_i \mathbf{Q}_i \,\middle|\, \sum_{i \in \mathcal{I}} \alpha_i = 1 \text{ and } \forall\ i \in \mathcal{I},\ \mathbf{Q}_i \in \mathcal{Q} \wedge \alpha_i \geq 0 \right\}$.

Notations

Recall

$$\begin{cases} x_i = \min\limits_{l \in \mathcal{N}(i)} x_l, & 1 \le i \le n_1, \\ x_j = \max\limits_{l \in \mathcal{N}(j)} x_l, & n_1 < j \le n_1 + n_2, \\ x_k = \mathbf{q}_k^{\mathsf{T}} \mathbf{x} + b_k, & n_1 + n_2 < k \le n. \end{cases}$$

- $\delta_{i,j}$ is 1 if $i = j$ and 0 otherwise.
- $\mathbf{e}_i \coloneqq [\delta_{i,1}, \ldots, \delta_{i,n}]^{\mathsf{T}},\ i \in [n]$.
- $\mathcal{Q} \coloneqq \left\{ [\mathbf{e}_{\ell_1}, \ldots, \mathbf{e}_{\ell_{n_1+n_2}}, \mathbf{q}_{n_1+n_2+1}, \ldots, \mathbf{q}_n]^{\mathsf{T}} \,\middle|\, \ell_j \in \mathcal{N}(j) \text{ for } j \in [n_1+n_2] \right\}$.
- $\mathbf{conv}(\mathcal{Q}) \coloneqq \left\{ \sum_{i \in \mathcal{I}} \alpha_i \mathbf{Q}_i \,\middle|\, \sum_{i \in \mathcal{I}} \alpha_i = 1 \text{ and } \forall\ i \in \mathcal{I},\ \mathbf{Q}_i \in \mathcal{Q} \wedge \alpha_i \ge 0 \right\}$.

Notations

Recall

$$\begin{cases} x_i = \min\limits_{l \in \mathcal{N}(i)} x_l, & 1 \leq i \leq n_1, \\ x_j = \max\limits_{l \in \mathcal{N}(j)} x_l, & n_1 < j \leq n_1 + n_2, \\ x_k = \mathbf{q}_k^\mathsf{T} \mathbf{x} + b_k, & n_1 + n_2 < k \leq n. \end{cases}$$

- $\delta_{i,j}$ is 1 if $i = j$ and 0 otherwise.
- $\mathbf{e}_i \coloneqq [\delta_{i,1}, \ldots, \delta_{i,n}]^\mathsf{T},\ i \in [n]$.
- $\mathcal{Q} \coloneqq \left\{ [\mathbf{e}_{\ell_1}, \ldots, \mathbf{e}_{\ell_{n_1+n_2}}, \mathbf{q}_{n_1+n_2+1}, \ldots, \mathbf{q}_n]^\mathsf{T} \,\middle|\, \ell_j \in \mathcal{N}(j) \text{ for } j \in [n_1 + n_2] \right\}$.
- $\mathbf{conv}(\mathcal{Q}) \coloneqq \left\{ \sum_{i \in \mathcal{I}} \alpha_i \mathbf{Q}_i \mid \sum_{i \in \mathcal{I}} \alpha_i = 1 \text{ and } \forall\ i \in \mathcal{I},\ \mathbf{Q}_i \in \mathcal{Q} \wedge \alpha_i \geq 0 \right\}$.

Restrictive conditions

Condition C1 (Halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} \mathbf{Q}^m = \mathbf{0}_{n\times n}$.

Condition C1+ (Absolutely halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} |\mathbf{Q}|^m = \mathbf{0}_{n\times n}$.

Condition C2 (Non-negative coefficients)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k \geq 0$ and $b_k \geq 0$.

Condition C3 (Sum upto 1)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k^\top \mathbf{1} + b_k \leq 1$.

Condition C4 (One-type)

Either $n_1 = 0$ or $n_2 = 0$.

Restrictive conditions

Condition C1 (Halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} \mathbf{Q}^m = \mathbf{0}_{n\times n}$.

Condition C1+ (Absolutely halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} |\mathbf{Q}|^m = \mathbf{0}_{n\times n}$.

Condition C2 (Non-negative coefficients)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k \geq 0$ and $b_k \geq 0$.

Condition C3 (Sum upto 1)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k^\top \mathbf{1} + b_k \leq 1$.

Condition C4 (One-type)

Either $n_1 = 0$ or $n_2 = 0$.

Restrictive conditions

Condition C1 (Halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} \mathbf{Q}^m = \mathbf{0}_{n\times n}$.

Condition C1+ (Absolutely halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} |\mathbf{Q}|^m = \mathbf{0}_{n\times n}$.

Condition C2 (Non-negative coefficients)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k \geq 0$ and $b_k \geq 0$.

Condition C3 (Sum upto 1)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k^\top \mathbf{1} + b_k \leq 1$.

Condition C4 (One-type)

Either $n_1 = 0$ or $n_2 = 0$.

Restrictive conditions

Condition C1 (Halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} \mathbf{Q}^m = \mathbf{0}_{n\times n}$.

Condition C1+ (Absolutely halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} |\mathbf{Q}|^m = \mathbf{0}_{n\times n}$.

Condition C2 (Non-negative coefficients)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k \geq 0$ and $b_k \geq 0$.

Condition C3 (Sum upto 1)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k^\top \mathbf{1} + b_k \leq 1$.

Condition C4 (One-type)

Either $n_1 = 0$ or $n_2 = 0$.

Restrictive conditions

Condition C1 (Halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} \mathbf{Q}^m = \mathbf{0}_{n\times n}$.

Condition C1+ (Absolutely halting)

For all $\mathbf{Q} \in \mathbf{conv}(\mathcal{Q})$, $\lim_{m\to\infty} |\mathbf{Q}|^m = \mathbf{0}_{n\times n}$.

Condition C2 (Non-negative coefficients)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k \geq 0$ and $b_k \geq 0$.

Condition C3 (Sum upto 1)

For all $n_1 + n_2 < k \leq n$, we have that $\mathbf{q}_k^\top \mathbf{1} + b_k \leq 1$.

Condition C4 (One-type)

Either $n_1 = 0$ or $n_2 = 0$.

Problem subclasses

- Well-studied problems:
  - Markov decision process (MDP): $\{C2, C3, C4\}$
  - Simple stochastic game (SSG): $\{C2, C3\}$
  - Branching process: $\{C2\}$
  - Halting (or absolutely halting): $\{C1\}$ (or $\{C1+\}$)
- Motivating examples:
  - Verifying neural networks: $\varnothing$
  - Capital preservation: $\{C1, C2\}$ or $\{C1+, C2\}$
  - Evolution in ecosystem: $\{C1, C4\}$

Problem subclasses

- Well-studied problems:
  - Markov decision process (MDP): $\{C2, C3, C4\}$
  - Simple stochastic game (SSG): $\{C2, C3\}$
  - Branching process: $\{C2\}$
  - Halting (or absolutely halting): $\{C1\}$ (or $\{C1+\}$)
- Motivating examples:
  - Verifying neural networks: $\varnothing$
  - Capital preservation: $\{C1, C2\}$ or $\{C1+, C2\}$
  - Evolution in ecosystem: $\{C1, C4\}$

Problem subclasses

- Well-studied problems:
  - Markov decision process (MDP): $\{C2, C3, C4\}$
  - Simple stochastic game (SSG): $\{C2, C3\}$
  - Branching process: $\{C2\}$
  - Halting (or absolutely halting): $\{C1\}$ (or $\{C1+\}$)
- Motivating examples:
  - Verifying neural networks: $\varnothing$
  - Capital preservation: $\{C1, C2\}$ or $\{C1+, C2\}$
  - Evolution in ecosystem: $\{C1, C4\}$

Problem subclasses

- Well-studied problems:
  - Markov decision process (MDP): $\{C2, C3, C4\}$
  - Simple stochastic game (SSG): $\{C2, C3\}$
  - Branching process: $\{C2\}$
  - Halting (or absolutely halting): $\{C1\}$ (or $\{C1+\}$)
- Motivating examples:
  - Verifying neural networks: $\emptyset$
  - Capital preservation: $\{C1, C2\}$ or $\{C1+, C2\}$
  - Evolution in ecosystem: $\{C1, C4\}$

Problem subclasses

- Well-studied problems:
  - Markov decision process (MDP): $\{C2, C3, C4\}$
  - Simple stochastic game (SSG): $\{C2, C3\}$
  - Branching process: $\{C2\}$
  - Halting (or absolutely halting): $\{C1\}$ (or $\{C1+\}$)
- Motivating examples:
  - Verifying neural networks: $\emptyset$
  - Capital preservation: $\{C1, C2\}$ or $\{C1+, C2\}$
  - Evolution in ecosystem: $\{C1, C4\}$

Problem subclasses

- Well-studied problems:
  - Markov decision process (MDP): $\{C2, C3, C4\}$
  - Simple stochastic game (SSG): $\{C2, C3\}$
  - Branching process: $\{C2\}$
  - Halting (or absolutely halting): $\{C1\}$ (or $\{C1+\}$)
- Motivating examples:
  - Verifying neural networks: $\varnothing$
  - Capital preservation: $\{C1, C2\}$ or $\{C1+, C2\}$
  - Evolution in ecosystem: $\{C1, C4\}$

Problem subclasses

- Well-studied problems:
  - Markov decision process (MDP): $\{C2, C3, C4\}$
  - Simple stochastic game (SSG): $\{C2, C3\}$
  - Branching process: $\{C2\}$
  - Halting (or absolutely halting): $\{C1\}$ (or $\{C1+\}$)
- Motivating examples:
  - Verifying neural networks: $\varnothing$
  - Capital preservation: $\{C1, C2\}$ or $\{C1+, C2\}$
  - Evolution in ecosystem: $\{C1, C4\}$

Problem subclasses

- Well-studied problems:
  - Markov decision process (MDP): $\{C2, C3, C4\}$
  - Simple stochastic game (SSG): $\{C2, C3\}$
  - Branching process: $\{C2\}$
  - Halting (or absolutely halting): $\{C1\}$ (or $\{C1+\}$)
- Motivating examples:
  - Verifying neural networks: $\varnothing$
  - Capital preservation: $\{C1, C2\}$ or $\{C1+, C2\}$
  - Evolution in ecosystem: $\{C1, C4\}$

Problem subclasses

- Well-studied problems:
  - Markov decision process (MDP): $\{C2, C3, C4\}$
  - Simple stochastic game (SSG): $\{C2, C3\}$
  - Branching process: $\{C2\}$
  - Halting (or absolutely halting): $\{C1\}$ (or $\{C1+\}$)
- Motivating examples:
  - Verifying neural networks: $\varnothing$
  - Capital preservation: $\{C1, C2\}$ or $\{C1+, C2\}$
  - Evolution in ecosystem: $\{C1, C4\}$

Outline

1. Introduction
   - Stochastic games
   - Optimization problem
2. Motivating examples and restrictive conditions
   - Motivating examples
   - Restrictive conditions
3. Complexity
   - Complexity of the decision problem
   - Complexity of checking the conditions
4. Algorithms
   - Classic algorithms for halting SSGs
   - Algorithms for absolutely halting subclass
   - Algorithms for halting subclass

Known complexity results

Basic fact. If $X \subseteq Y$ represents two subsets of conditions, then the decision problem under X is no easier than the decision problem under Y.

$\varnothing$	NP-complete
$\{C1, C2, C3\}$, $\{C2, C3\}$	UP $\cap$ coUP (SSG-hard)
$\{C1, C2, C3, C4\}$, $\{C2, C3, C4\}$	PTIME

Table 1: The known complexity of subclasses of decision problems.[5]

[5] Condon, "The complexity of stochastic games".

Known complexity results

Basic fact. If $X \subseteq Y$ represents two subsets of conditions, then the decision problem under X is no easier than the decision problem under Y.

$\varnothing$	NP-complete
$\{C1, C2, C3\}$, $\{C2, C3\}$	UP $\cap$ coUP (SSG-hard)
$\{C1, C2, C3, C4\}$, $\{C2, C3, C4\}$	PTIME

Table 1: The known complexity of subclasses of decision problems.[5]

[5] Condon, "The complexity of stochastic games".

Complexity lower bounds

Lemma 1

The decision problem under $\{C2, C4\}$ *is NP-hard.*

Lemma 2

The decision problem under $\{C3, C4\}$ *is NP-hard.*

$\{C2, C4\}$, $\{C3, C4\}$, $\{C2\}$, $\{C3\}$, $\{C4\}$, $\varnothing$	NP-complete
$\{C1, C2, C3\}$, $\{C2, C3\}$	UP $\cap$ coUP (SSG-hard)
$\{C1, C2, C3, C4\}$, $\{C2, C3, C4\}$	PTIME

Table 2: The complexity of subclasses of decision problems.

Complexity lower bounds

Lemma 1

The decision problem under $\{\mathrm{C2}, \mathrm{C4}\}$ *is NP-hard.*

Lemma 2

The decision problem under $\{\mathrm{C3}, \mathrm{C4}\}$ *is NP-hard.*

$\{\mathrm{C2}, \mathrm{C4}\}$, $\{\mathrm{C3}, \mathrm{C4}\}$, $\{\mathrm{C2}\}$, $\{\mathrm{C3}\}$, $\{\mathrm{C4}\}$, $\varnothing$	NP-complete
$\{\mathrm{C1}, \mathrm{C2}, \mathrm{C3}\}$, $\{\mathrm{C2}, \mathrm{C3}\}$	UP $\cap$ coUP (SSG-hard)
$\{\mathrm{C1}, \mathrm{C2}, \mathrm{C3}, \mathrm{C4}\}$, $\{\mathrm{C2}, \mathrm{C3}, \mathrm{C4}\}$	PTIME

Table 2: The complexity of subclasses of decision problems.

Complexity lower bounds

Lemma 1

The decision problem under $\{\mathrm{C2}, \mathrm{C4}\}$ *is NP-hard.*

Lemma 2

The decision problem under $\{\mathrm{C3}, \mathrm{C4}\}$ *is NP-hard.*

$\{\mathrm{C2}, \mathrm{C4}\}$, $\{\mathrm{C3}, \mathrm{C4}\}$, $\{\mathrm{C2}\}$, $\{\mathrm{C3}\}$, $\{\mathrm{C4}\}$, $\varnothing$	NP-complete
$\{\mathrm{C1}, \mathrm{C2}, \mathrm{C3}\}$, $\{\mathrm{C2}, \mathrm{C3}\}$	UP $\cap$ coUP (SSG-hard)
$\{\mathrm{C1}, \mathrm{C2}, \mathrm{C3}, \mathrm{C4}\}$, $\{\mathrm{C2}, \mathrm{C3}, \mathrm{C4}\}$	PTIME

Table 2: The complexity of subclasses of decision problems.

Complexity upper bounds

Lemma 3

The decision problem under $\{C1, C2, C4\}$ *is in PTIME.*

$\{C2, C4\}$, $\{C3, C4\}$, $\{C2\}$, $\{C3\}$, $\{C4\}$, $\varnothing$	NP-complete
$\{C1, C2, C3\}$, $\{C2, C3\}$	UP $\cap$ coUP (SSG-hard)
$\{C1, C2, C3, C4\}$, $\{C2, C3, C4\}$, $\{C1, C2, C4\}$	PTIME

Table 3: The complexity of subclasses of decision problems.

Complexity upper bounds

Lemma 3

The decision problem under $\{C1, C2, C4\}$ *is in PTIME.*

$\{C2, C4\}$, $\{C3, C4\}$, $\{C2\}$, $\{C3\}$, $\{C4\}$, $\varnothing$	NP-complete
$\{C1, C2, C3\}$, $\{C2, C3\}$	UP $\cap$ coUP (SSG-hard)
$\{C1, C2, C3, C4\}$, $\{C2, C3, C4\}$, $\{C1, C2, C4\}$	PTIME

Table 3: The complexity of subclasses of decision problems.

Complexity upper bounds

Lemma 4

The decision problem under $\{C1\}$ *is in UP* $\cap$ *coUP.*

The classic proof for SSGs follow from minimax theorem[6]:

- For UP, one guesses MIN strategy;
- For coUP, one guesses MAX strategy.

However, this classic proof does not work for $\{C1\}$, since minimax theorem breaks here.

Proof idea.

We show that *under* $\{C1\}$, *the Equation* (1) *has a unique solution*, which can be used as the unique certificate for UP or for coUP. □

[6] Condon, "The complexity of stochastic games".

Complexity upper bounds

Lemma 4

The decision problem under $\{C1\}$ *is in UP* $\cap$ *coUP.*

The classic proof for SSGs follow from minimax theorem[6]:

- For UP, one guesses $\textsc{Min}$ strategy;
- For coUP, one guesses $\textsc{Max}$ strategy.

However, this classic proof does not work for $\{C1\}$, since minimax theorem breaks here.

Proof idea.

We show that *under* $\{C1\}$, *the Equation* (1) *has a unique solution*, which can be used as the unique certificate for UP or for coUP. $\square$

[6] Condon, "The complexity of stochastic games".

Complexity upper bounds

Lemma 4

The decision problem under $\{\mathrm{C1}\}$ *is in UP* $\cap$ *coUP.*

The classic proof for SSGs follow from minimax theorem[6]:

- For UP, one guesses $\textsc{Min}$ strategy;
- For coUP, one guesses $\textsc{Max}$ strategy.

However, this classic proof does not work for $\{\mathrm{C1}\}$, since minimax theorem breaks here.

Proof idea.

We show that *under* $\{\mathrm{C1}\}$, *the Equation* (1) *has a unique solution*, which can be used as the unique certificate for UP or for coUP. □

[6] Condon, "The complexity of stochastic games".

Complexity upper bounds

Lemma 4

The decision problem under $\{\mathrm{C1}\}$ *is in UP* $\cap$ *coUP.*

The classic proof for SSGs follow from minimax theorem[6]:

- For UP, one guesses $\textsc{Min}$ strategy;
- For coUP, one guesses $\textsc{Max}$ strategy.

However, this classic proof does not work for $\{\mathrm{C1}\}$, since minimax theorem breaks here.

Proof idea.

We show that *under* $\{\mathrm{C1}\}$, *the Equation* (1) *has a unique solution*, which can be used as the unique certificate for UP or for coUP. $\square$

[6] Condon, "The complexity of stochastic games".

Equivalence between subclasses

Lemma 5

The decision problems under $\{\mathrm{C1}\}$ and $\{\mathrm{C1}, \mathrm{C3}, \mathrm{C4}\}$ are equivalent.
The decision problems under $\{\mathrm{C1+}\}$ and $\{\mathrm{C1+}, \mathrm{C3}, \mathrm{C4}\}$ are equivalent.

{C2,C4}, {C3,C4}, {C2}, {C3}, {C4}, $\varnothing$	NP-complete
{C1,C3,C4}, {C1,C3}, {C1,C4}, {C1}	UP $\cap$ coUP (SSG-hard)
{C1+,C3,C4}, {C1+,C3}, {C1+,C4}, {C1+}	
{C1,C2}	
{C1,C2,C3}, {C2,C3}	
{C1,C2,C3,C4}, {C2,C3,C4}, {C1, C2, C4}	PTIME

Table 4: The complexity of subclasses of decision problems.

Equivalence between subclasses

Lemma 5

The decision problems under $\{C1\}$ *and* $\{C1, C3, C4\}$ *are equivalent.*
The decision problems under $\{C1+\}$ *and* $\{C1+, C3, C4\}$ *are equivalent.*

{C2,C4}, {C3,C4}, {C2}, {C3}, {C4}, ∅	NP-complete
{C1,C3,C4}, {C1,C3}, {C1,C4}, {C1} {C1+,C3,C4}, {C1+,C3}, {C1+,C4}, {C1+} {C1,C2} {C1,C2,C3}, {C2,C3}	UP ∩ coUP (SSG-hard)
{C1,C2,C3,C4}, {C2,C3,C4}, {C1, C2, C4}	PTIME

Table 4: The complexity of subclasses of decision problems.

Equivalence between subclasses

Lemma 6

The decision problems under $\{C1, C2\}$ *and* $\{C1, C2, C3\}$ *are equivalent.*[a]

[a] $\{C1, C2\}$ is trivially equivalent to $\{C1+, C2\}$.

Theorem 1

$\{C2,C4\}$, $\{C3,C4\}$, $\{C2\}$, $\{C3\}$, $\{C4\}$, $\varnothing$	*NP-complete*
$\{C1,C3,C4\}$, $\{C1,C3\}$, $\{C1,C4\}$, $\{C1\}$ $\{C1+,C3,C4\}$, $\{C1+,C3\}$, $\{C1+,C4\}$, $\{C1+\}$ $\{C1,C2,C3\}$, $\{C2,C3\}$, $\{C1,C2\}$	*UP ∩ coUP (SSG-hard)*
$\{C1,C2,C3,C4\}$, $\{C2,C3,C4\}$, $\{C1, C2, C4\}$	*PTIME*

Table 5: *The complexity of decision problems under all subsets of conditions.*

We obtain new problem classes in UP ∩ coUP (while generalizing SSGs):
(i) $\{C1+\}$; *(ii)* $\{C1\}$.

Equivalence between subclasses

Lemma 6

The decision problems under $\{\mathrm{C1}, \mathrm{C2}\}$ *and* $\{\mathrm{C1}, \mathrm{C2}, \mathrm{C3}\}$ *are equivalent.*[a]

[a] $\{\mathrm{C1}, \mathrm{C2}\}$ is trivially equivalent to $\{\mathrm{C1}+, \mathrm{C2}\}$.

Theorem 1

{C2,C4}, {C3,C4}, {C2}, {C3}, {C4}, ∅	*NP-complete*
{C1,C3,C4}, {C1,C3}, {C1,C4}, {C1} {C1+,C3,C4}, {C1+,C3}, {C1+,C4}, {C1+} {C1,C2,C3}, {C2,C3}, {C1,C2}	*UP ∩ coUP (SSG-hard)*
{C1,C2,C3,C4}, {C2,C3,C4}, {C1, C2, C4}	*PTIME*

Table 5: *The complexity of decision problems under all subsets of conditions.*

We obtain new problem classes in UP ∩ coUP (while generalizing SSGs):
(i) $\{\mathrm{C1}+\}$*;* *(ii)* $\{\mathrm{C1}\}$.

Equivalence between subclasses

Lemma 6

The decision problems under $\{\mathrm{C1}, \mathrm{C2}\}$ *and* $\{\mathrm{C1}, \mathrm{C2}, \mathrm{C3}\}$ *are equivalent.*[a]

[a] $\{\mathrm{C1}, \mathrm{C2}\}$ is trivially equivalent to $\{\mathrm{C1+}, \mathrm{C2}\}$.

Theorem 1

{C2,C4}, {C3,C4}, {C2}, {C3}, {C4}, ∅	*NP-complete*
{C1,C3,C4}, {C1,C3}, {C1,C4}, {C1} {C1+,C3,C4}, {C1+,C3}, {C1+,C4}, {C1+} {C1,C2,C3}, {C2,C3}, {C1,C2}	*UP ∩ coUP (SSG-hard)*
{C1,C2,C3,C4}, {C2,C3,C4}, {C1, C2, C4}	*PTIME*

Table 5: *The complexity of decision problems under all subsets of conditions.*

We obtain new problem classes in UP ∩ coUP (while generalizing SSGs):
(i) $\{\mathrm{C1+}\}$*;* *(ii)* $\{\mathrm{C1}\}$.

Outline

1. Introduction
   - Stochastic games
   - Optimization problem
2. Motivating examples and restrictive conditions
   - Motivating examples
   - Restrictive conditions
3. Complexity
   - Complexity of the decision problem
   - Complexity of checking the conditions
4. Algorithms
   - Classic algorithms for halting SSGs
   - Algorithms for absolutely halting subclass
   - Algorithms for halting subclass

Condition checking problem

Definition 2 (Condition checking problem)

Given Equation (1) and a subset of conditions, determine whether all the conditions are satisfied.

Basic fact. All of Conditions C2 to C4 can be trivially checked in linear time.

The non-trivial condition checking problems are the ones including Condition C1 or Condition C1+.

If $\{C1\} \subseteq X \subseteq Y$ (or $\{C1+\} \subseteq X \subseteq Y$) represents two subsets of conditions, then the condition checking problem for X is no easier than the condition checking problem for Y.

{C1,C2,C3}, {C1,C2,C3,C4}	PTIME

Table 6: The known complexity of condition checking problems.

Condition checking problem

Definition 2 (Condition checking problem)

Given Equation (1) and a subset of conditions, determine whether all the conditions are satisfied.

Basic fact. All of Conditions C2 to C4 can be trivially checked in linear time.

The non-trivial condition checking problems are the ones including Condition C1 or Condition C1+.

If $\{C1\} \subseteq X \subseteq Y$ (or $\{C1+\} \subseteq X \subseteq Y$) represents two subsets of conditions, then the condition checking problem for X is no easier than the condition checking problem for Y.

{C1,C2,C3}, {C1,C2,C3,C4}	PTIME

Table 6: The known complexity of condition checking problems.

Condition checking problem

Definition 2 (Condition checking problem)

Given Equation (1) and a subset of conditions, determine whether all the conditions are satisfied.

Basic fact. All of Conditions C2 to C4 can be trivially checked in linear time. The non-trivial condition checking problems are the ones including Condition C1 or Condition C1+.

If $\{C1\} \subseteq X \subseteq Y$ (or $\{C1+\} \subseteq X \subseteq Y$) represents two subsets of conditions, then the condition checking problem for X is no easier than the condition checking problem for Y.

{C1,C2,C3}, {C1,C2,C3,C4}	PTIME

Table 6: The known complexity of condition checking problems.

Condition checking problem

Definition 2 (Condition checking problem)

Given Equation (1) and a subset of conditions, determine whether all the conditions are satisfied.

Basic fact. All of Conditions C2 to C4 can be trivially checked in linear time. The non-trivial condition checking problems are the ones including Condition C1 or Condition C1+.

If $\{C1\} \subseteq X \subseteq Y$ (or $\{C1+\} \subseteq X \subseteq Y$) represents two subsets of conditions, then the condition checking problem for X is no easier than the condition checking problem for Y.

{C1,C2,C3}, {C1,C2,C3,C4}	PTIME

Table 6: The known complexity of condition checking problems.

Condition checking problem

Definition 2 (Condition checking problem)

Given Equation (1) and a subset of conditions, determine whether all the conditions are satisfied.

Basic fact. All of Conditions C2 to C4 can be trivially checked in linear time. The non-trivial condition checking problems are the ones including Condition C1 or Condition C1+.
If $\{C1\} \subseteq X \subseteq Y$ (or $\{C1+\} \subseteq X \subseteq Y$) represents two subsets of conditions, then the condition checking problem for X is no easier than the condition checking problem for Y.

{C1,C2,C3}, {C1,C2,C3,C4}	PTIME

Table 6: The known complexity of condition checking problems.

Checking the absolutely halting condition

Lemma 7

The condition checking problems for $\{\mathrm{C1+}\}$ *and* $\{\mathrm{C1}, \mathrm{C2}\}$ *are in PTIME.*

$\{\mathrm{C1+}\}, \cdots, \{\mathrm{C1,C2}\}, \cdots, \{\mathrm{C1,C2,C3}\}, \{\mathrm{C1,C2,C3,C4}\}$	PTIME

Table 7: The complexity of condition checking problems.

Checking the absolutely halting condition

Lemma 7

The condition checking problems for $\{\text{C1}+\}$ *and* $\{\text{C1}, \text{C2}\}$ *are in PTIME.*

$\{\text{C1}+\}, \cdots, \{\text{C1,C2}\}, \cdots, \{\text{C1,C2,C3}\}, \{\text{C1,C2,C3,C4}\}$	PTIME

Table 7: The complexity of condition checking problems.

Checking the halting condition

Lemma 8

The condition checking problem for {C1,C3,C4} *is coNP-hard.*

Lemma 9

The condition checking problem for {C1} *is in coNP.*

Theorem 2

{C1}, {C1,C3}, {C1,C4}, {C1,C3,C4}	*coNP-comp.*
{C1+}, $\cdots$, {C1,C2}, $\cdots$, {C1,C2,C3}, {C1,C2,C3,C4}	*PTIME*

Table 8: *The complexity of all the condition checking problems.*

Checking the halting condition

Lemma 8

The condition checking problem for {C1,C3,C4} *is coNP-hard.*

Lemma 9

The condition checking problem for {C1} *is in coNP.*

Theorem 2

{C1}, {C1,C3}, {C1,C4}, {C1,C3,C4}	*coNP-comp.*
{C1+}, $\cdots$, {C1,C2}, $\cdots$, {C1,C2,C3}, {C1,C2,C3,C4}	*PTIME*

Table 8: *The complexity of all the condition checking problems.*

Checking the halting condition

Lemma 8

The condition checking problem for {C1,C3,C4} *is coNP-hard.*

Lemma 9

The condition checking problem for {C1} *is in coNP.*

Theorem 2

{C1}, {C1,C3}, {C1,C4}, {C1,C3,C4}	*coNP-comp.*
{C1+}, $\cdots$, {C1,C2}, $\cdots$, {C1,C2,C3}, {C1,C2,C3,C4}	*PTIME*

Table 8: *The complexity of all the condition checking problems.*

Outline

1. Introduction
   - Stochastic games
   - Optimization problem
2. Motivating examples and restrictive conditions
   - Motivating examples
   - Restrictive conditions
3. Complexity
   - Complexity of the decision problem
   - Complexity of checking the conditions
4. Algorithms
   - Classic algorithms for halting SSGs
   - Algorithms for absolutely halting subclass
   - Algorithms for halting subclass

Notations

Recall

$$\begin{cases} x_i = \min\limits_{l \in \mathcal{N}(i)} x_l, & 1 \leq i \leq n_1, \\ x_j = \max\limits_{l \in \mathcal{N}(j)} x_l, & n_1 < j \leq n_1 + n_2, \\ x_k = \mathbf{q}_k^{\mathsf{T}} \mathbf{x} + b_k, & n_1 + n_2 < k \leq n. \end{cases}$$

Denote

$$\mathcal{Q}_{\min} = \left\{ [\mathbf{e}_{\ell_1}, \cdots, \mathbf{e}_{\ell_{n_1}}]^{\mathsf{T}} \,\middle|\, \ell_i \in \mathcal{J}(i) \text{ for } 1 \leq i \leq n_1 \right\},$$

$$\mathcal{Q}_{\max} = \left\{ [\mathbf{e}_{\ell_{n_1+1}}, \cdots, \mathbf{e}_{\ell_{n_1+n_2}}]^{\mathsf{T}} \,\middle|\, \ell_j \in \mathcal{J}(j) \text{ for } n_1 < j \leq n_2 \right\},$$

$$\mathbf{Q}_{\mathrm{aff}} = [\mathbf{q}_{n_1+n_2+1}, \ldots, \mathbf{q}_n]^{\mathsf{T}}.$$

Then,

$$\mathcal{Q} = \left\{ \begin{bmatrix} \mathbf{Q}_{\min} \\ \mathbf{Q}_{\max} \\ \mathbf{Q}_{\mathrm{aff}} \end{bmatrix} \,\middle|\, \mathbf{Q}_{\min} \in \mathcal{Q}_{\min} \text{ and } \mathbf{Q}_{\max} \in \mathcal{Q}_{\max} \right\}.$$

Notations

Recall

$$\begin{cases} x_i = \min_{l \in \mathcal{N}(i)} x_l, & 1 \le i \le n_1, \\ x_j = \max_{l \in \mathcal{N}(j)} x_l, & n_1 < j \le n_1 + n_2, \\ x_k = \mathbf{q}_k^\mathsf{T} \mathbf{x} + b_k, & n_1 + n_2 < k \le n. \end{cases}$$

Denote

$$\mathcal{Q}_{\min} = \left\{ [\mathbf{e}_{\ell_1}, \cdots, \mathbf{e}_{\ell_{n_1}}]^\mathsf{T} \middle| \ \ell_i \in \mathcal{J}(i) \text{ for } 1 \le i \le n_1 \right\},$$

$$\mathcal{Q}_{\max} = \left\{ [\mathbf{e}_{\ell_{n_1+1}}, \cdots, \mathbf{e}_{\ell_{n_1+n_2}}]^\mathsf{T} \middle| \ \ell_j \in \mathcal{J}(j) \text{ for } n_1 < j \le n_2 \right\},$$

$$\mathbf{Q}_{\text{aff}} = [\mathbf{q}_{n_1+n_2+1}, \ldots, \mathbf{q}_n]^\mathsf{T}.$$

Then,

$$\mathcal{Q} = \left\{ \begin{bmatrix} \mathbf{Q}_{\min} \\ \mathbf{Q}_{\max} \\ \mathbf{Q}_{\text{aff}} \end{bmatrix} \middle| \ \mathbf{Q}_{\min} \in \mathcal{Q}_{\min} \text{ and } \mathbf{Q}_{\max} \in \mathcal{Q}_{\max} \right\}.$$

Notations

Recall

$$\begin{cases} x_i = \min\limits_{l \in \mathcal{N}(i)} \; x_l, & 1 \le i \le n_1, \\ x_j = \max\limits_{l \in \mathcal{N}(j)} \; x_l, & n_1 < j \le n_1 + n_2, \\ x_k = \mathbf{q}_k^\mathsf{T} \mathbf{x} + b_k, & n_1 + n_2 < k \le n. \end{cases}$$

Denote

$$\begin{aligned} \mathcal{Q}_{\min} &= \left\{ \left[\mathbf{e}_{\ell_1}, \cdots, \mathbf{e}_{\ell_{n_1}}\right]^\mathsf{T} \middle| \; \ell_i \in \mathcal{J}(i) \text{ for } 1 \le i \le n_1 \right\}, \\ \mathcal{Q}_{\max} &= \left\{ \left[\mathbf{e}_{\ell_{n_1+1}}, \cdots, \mathbf{e}_{\ell_{n_1+n_2}}\right]^\mathsf{T} \middle| \; \ell_j \in \mathcal{J}(j) \text{ for } n_1 < j \le n_2 \right\}, \\ \mathbf{Q}_{\text{aff}} &= \left[\mathbf{q}_{n_1+n_2+1}, \ldots, \mathbf{q}_n\right]^\mathsf{T}. \end{aligned}$$

Then,

$$\mathcal{Q} = \left\{ \begin{bmatrix} \mathbf{Q}_{\min} \\ \mathbf{Q}_{\max} \\ \mathbf{Q}_{\text{aff}} \end{bmatrix} \middle| \; \mathbf{Q}_{\min} \in \mathcal{Q}_{\min} \text{ and } \mathbf{Q}_{\max} \in \mathcal{Q}_{\max} \right\}.$$

Policy iteration

For all max policy $\mathbf{Q} \in \mathcal{Q}_{\max}$, define its value estimate $\mathbf{x}_{\max}(\mathbf{Q}) \in \mathbb{R}^n$ s.t.

$$\begin{cases} x_i = \min\limits_{l \in \mathcal{J}(i)} x_l, & 1 \le i \le n_1\,, \\ \left[x_{n_1+1}, \cdots, x_{n_1+n_2}\right]^{\mathsf{T}} = \mathbf{Q}\mathbf{x}\,, & \\ x_k = \mathbf{q}_k^{\mathsf{T}}\mathbf{x} + b_k, & n_1 + n_2 < k \le n\,. \end{cases} \tag{2}$$

Further, define the max policy extraction $\pi_{\max}(\mathbf{x}) \colon \mathbb{R}^n \to \mathcal{Q}_{\max}$ s.t.

$$\pi_{\max}(\mathbf{x}) \cdot \mathbf{x} = \max_{\mathbf{Q}' \in \mathcal{Q}_{\max}} \mathbf{Q}' \cdot \mathbf{x}\,.$$

Algorithm 1 Policy Iteration (PI) updating max policies

Require: $\mathbf{Q}^{(0)} \in \mathcal{Q}_{\max}$

for $t = 1, 2, \cdots$ **do**

$\quad \mathbf{x}^{(t)} = \mathbf{x}_{\max}(\mathbf{Q}^{(t-1)})$

$\quad \mathbf{Q}^{(t)} = \pi_{\max}(\mathbf{x}^{(t)})$

end for

Value iteration

Define the value iteration operator $\upsilon(\mathbf{x})\colon \mathbb{R}^n \to \mathbb{R}^n$ s.t.

$$\begin{cases} \upsilon_i(\mathbf{x}) = \min\limits_{l \in \mathcal{J}(i)} x_l, & 1 \le i \le n_1\,, \\ \upsilon_j(\mathbf{x}) = \max\limits_{l \in \mathcal{J}(j)} x_l, & n_1 < j \le n_1 + n_2\,, \\ \upsilon_k(\mathbf{x}) = \mathbf{q}_k^\mathsf{T}\mathbf{x} + b_k, & n_1 + n_2 < k \le n\,. \end{cases} \tag{3}$$

Algorithm 2 Value Iteration (VI)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^n$

for $t = 1, 2, \cdots$ **do**

$\quad \mathbf{x}^{(t)} = \upsilon\left(\mathbf{x}^{(t-1)}\right)$

end for

Summary: Algorithms for halting stochastic games

Proposition 1 ([7])

For Equation (1) *under Conditions* C1 *to* C3*:*

- PI *converges to the exact solution of Equation* (1) *in no more than* $\Pi_{i\in(n_1,n_1+n_2]} |\mathcal{N}(i)|$ *iterations;*
- *Initialized at* $\mathbf{0} \in \mathbb{R}^n$ *or at* $\mathbf{1} \in \mathbb{R}^n$, VI *converges linearly to the solution of Equation* (1) *at a ratio of*

$$\widetilde{\gamma} \triangleq \max\left\{ \left\| \mathbf{Q}^{(n)} \cdots \mathbf{Q}^{(1)} \right\|_\infty^{\frac{1}{n}} \mid \mathbf{Q}^{(i)} \in \mathcal{Q} \text{ for all } i \in [n] \right\} < 1.$$

[7] Anne Condon. "On Algorithms for Simple Stochastic Games.". In: *Advances in computational complexity theory* 13 (1990), pp. 51–72.

Summary: Algorithms for halting stochastic games

Proposition 1 ([7])

For Equation (1) *under Conditions* C1 *to* C3*:*

- PI *converges to the exact solution of Equation* (1) *in no more than* $\Pi_{i \in (n_1, n_1+n_2]} |\mathcal{N}(i)|$ *iterations;*
- *Initialized at* $\mathbf{0} \in \mathbb{R}^n$ *or at* $\mathbf{1} \in \mathbb{R}^n$, VI *converges linearly to the solution of Equation* (1) *at a ratio of*

$$\tilde{\gamma} \triangleq \max \left\{ \left\| \mathbf{Q}^{(n)} \cdots \mathbf{Q}^{(1)} \right\|_{\infty}^{\frac{1}{n}} \mid \mathbf{Q}^{(i)} \in \mathcal{Q} \text{ for all } i \in [n] \right\} < 1.$$

[7] Condon, "On Algorithms for Simple Stochastic Games."

Summary: Algorithms for halting stochastic games

Proposition 1 ([7])

For Equation (1) *under Conditions* C1 *to* C3*:*

- PI *converges to the exact solution of Equation* (1) *in no more than* $\mathit{\Pi}_{i \in (n_1, n_1+n_2]} |\mathcal{N}(i)|$ *iterations;*
- *Initialized at* $\mathbf{0} \in \mathbb{R}^n$ *or at* $\mathbf{1} \in \mathbb{R}^n$, VI *converges linearly to the solution of Equation* (1) *at a ratio of*

$$\widetilde{\gamma} \triangleq \max\left\{ \left\| \mathbf{Q}^{(n)} \cdots \mathbf{Q}^{(1)} \right\|_{\infty}^{\frac{1}{n}} \mid \mathbf{Q}^{(i)} \in \mathcal{Q} \textit{ for all } i \in [n] \right\} < 1.$$

[7] Condon, "On Algorithms for Simple Stochastic Games."

Outline

1. Introduction
   - Stochastic games
   - Optimization problem
2. Motivating examples and restrictive conditions
   - Motivating examples
   - Restrictive conditions
3. Complexity
   - Complexity of the decision problem
   - Complexity of checking the conditions
4. Algorithms
   - Classic algorithms for halting SSGs
   - Algorithms for absolutely halting subclass
   - Algorithms for halting subclass

Policy iteration under Condition C1+

A counterexample satisfying Conditions C1+, C3 and C4:

$$\begin{cases} x_1 = \max\{x_4, x_5\}, \\ x_2 = \max\{x_6, x_7\}, \\ x_3 = \max\{x_8, x_9\}, \\ x_4 = -0.2x_2 + 0.2x_3 + 0.25, \\ x_5 = 0.3x_1 - 0.6x_3 + 0.25, \\ x_6 = -0.5x_1 + 0.25, \\ x_7 = 0.3x_1 + 0.1x_2 - 0.3x_3 + 0.25, \\ x_8 = -0.2x_1 + 0.4x_2 + 0.3, \\ x_9 = -0.7x_2 + 0.3x_3 + 0.3. \end{cases}$$

Initialized at

$$\mathbf{Q}_{\max}^{(0)} = [\mathbf{e}_4, \mathbf{e}_6, \mathbf{e}_8]^{\mathsf{T}},$$

PI gets

$$\mathbf{Q}_{\max}^{(1)} = [\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_8]^{\mathsf{T}},$$
$$\mathbf{Q}_{\max}^{(2)} = [\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_9]^{\mathsf{T}},$$
$$\mathbf{Q}_{\max}^{(3)} = [\mathbf{e}_5, \mathbf{e}_7, \mathbf{e}_9]^{\mathsf{T}},$$
$$\mathbf{Q}_{\max}^{(4)} = [\mathbf{e}_5, \mathbf{e}_7, \mathbf{e}_8]^{\mathsf{T}},$$
$$\mathbf{Q}_{\max}^{(5)} = [\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_8]^{\mathsf{T}},$$
$$\mathbf{Q}_{\max}^{(6)} = [\mathbf{e}_4, \mathbf{e}_6, \mathbf{e}_8]^{\mathsf{T}},$$
$$\cdots\cdots$$

and enters a loop.

Policy iteration under Condition C1+

A counterexample satisfying Conditions C1+, C3 and C4:

$$\begin{cases} x_1 = \max\{x_4, x_5\}, \\ x_2 = \max\{x_6, x_7\}, \\ x_3 = \max\{x_8, x_9\}, \\ x_4 = -0.2x_2 + 0.2x_3 + 0.25, \\ x_5 = 0.3x_1 - 0.6x_3 + 0.25, \\ x_6 = -0.5x_1 + 0.25, \\ x_7 = 0.3x_1 + 0.1x_2 - 0.3x_3 + 0.25, \\ x_8 = -0.2x_1 + 0.4x_2 + 0.3, \\ x_9 = -0.7x_2 + 0.3x_3 + 0.3. \end{cases}$$

Initialized at

$$\mathbf{Q}_{\max}^{(0)} = [\mathbf{e}_4, \mathbf{e}_6, \mathbf{e}_8]^\mathsf{T},$$

PI gets

$$\mathbf{Q}_{\max}^{(1)} = [\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_8]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(2)} = [\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_9]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(3)} = [\mathbf{e}_5, \mathbf{e}_7, \mathbf{e}_9]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(4)} = [\mathbf{e}_5, \mathbf{e}_7, \mathbf{e}_8]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(5)} = [\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_8]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(6)} = [\mathbf{e}_4, \mathbf{e}_6, \mathbf{e}_8]^\mathsf{T},$$

$\cdots\cdots$

and enters a loop.

Policy iteration under Condition C1+

A counterexample satisfying Conditions C1+, C3 and C4:

$$\begin{cases} x_1 = \max\{x_4, x_5\}, \\ x_2 = \max\{x_6, x_7\}, \\ x_3 = \max\{x_8, x_9\}, \\ x_4 = -0.2x_2 + 0.2x_3 + 0.25, \\ x_5 = 0.3x_1 - 0.6x_3 + 0.25, \\ x_6 = -0.5x_1 + 0.25, \\ x_7 = 0.3x_1 + 0.1x_2 - 0.3x_3 + 0.25, \\ x_8 = -0.2x_1 + 0.4x_2 + 0.3, \\ x_9 = -0.7x_2 + 0.3x_3 + 0.3. \end{cases}$$

Initialized at

$$\mathbf{Q}_{\max}^{(0)} = [\mathbf{e}_4, \mathbf{e}_6, \mathbf{e}_8]^\mathsf{T},$$

PI gets

$$\mathbf{Q}_{\max}^{(1)} = [\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_8]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(2)} = [\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_9]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(3)} = [\mathbf{e}_5, \mathbf{e}_7, \mathbf{e}_9]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(4)} = [\mathbf{e}_5, \mathbf{e}_7, \mathbf{e}_8]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(5)} = [\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_8]^\mathsf{T},$$
$$\mathbf{Q}_{\max}^{(6)} = [\mathbf{e}_4, \mathbf{e}_6, \mathbf{e}_8]^\mathsf{T},$$
$$\cdots\cdots$$

and enters a loop.

Value iteration under Condition C1+

Recall the ratio in the classic analysis of halting SSGs:

$$\widetilde{\gamma} = \max \left\{ \left\| \mathbf{Q}^{(n)} \cdots \mathbf{Q}^{(1)} \right\|_{\infty}^{\frac{1}{n}} \mid \mathbf{Q}^{(i)} \in \mathcal{Q} \text{ for all } i \in [n] \right\} .$$

However, for {C1+}, consider

$$\mathbf{Q} = \begin{bmatrix} 0.5 & 0.7 \\ 0 & 0.7 \end{bmatrix}$$

with spectral radius $\rho(\mathbf{Q}) = 0.7 < 1$, yet we have $\widetilde{\gamma}^2 = \left\| \mathbf{Q}^2 \mathbf{1} \right\|_{\infty} = 1.49 > 1$.

Lemma 10 (New analysis)

For Equation (1) *under Condition* C1+, *initialized at any point,* VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}) .$$

Value iteration under Condition C1+

Recall the ratio in the classic analysis of halting SSGs:

$$\widetilde{\gamma} = \max\left\{ \left\| \mathbf{Q}^{(n)} \cdots \mathbf{Q}^{(1)} \right\|_{\infty}^{\frac{1}{n}} \mid \mathbf{Q}^{(i)} \in \mathcal{Q} \text{ for all } i \in [n] \right\}.$$

However, for $\{\mathrm{C1}+\}$, consider

$$\mathbf{Q} = \begin{bmatrix} 0.5 & 0.7 \\ 0 & 0.7 \end{bmatrix}$$

with spectral radius $\rho(\mathbf{Q}) = 0.7 < 1$, yet we have
$\widetilde{\gamma}^2 = \left\| \mathbf{Q}^2 \mathbf{1} \right\|_{\infty} = 1.49 > 1$.

Lemma 10 (New analysis)

For Equation (1) under Condition C1+, initialized at any point, VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}).$$

Value iteration under Condition C1+

Recall the ratio in the classic analysis of halting SSGs:

$$\widetilde{\gamma} = \max\left\{ \left\| \mathbf{Q}^{(n)} \cdots \mathbf{Q}^{(1)} \right\|_{\infty}^{\frac{1}{n}} \mid \mathbf{Q}^{(i)} \in \mathcal{Q} \text{ for all } i \in [n] \right\} .$$

However, for $\{\mathrm{C1}+\}$, consider

$$\mathbf{Q} = \begin{bmatrix} 0.5 & 0.7 \\ 0 & 0.7 \end{bmatrix}$$

with spectral radius $\rho(\mathbf{Q}) = 0.7 < 1$, yet we have
$\widetilde{\gamma}^2 = \left\| \mathbf{Q}^2 \mathbf{1} \right\|_{\infty} = 1.49 > 1$.

Lemma 10 (New analysis)

For Equation (1) *under Condition* C1+, *initialized at any point,* VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}) .$$

Value iteration under Condition C1+

Recall the ratio in the classic analysis of halting SSGs:

$$\widetilde{\gamma} = \max \left\{ \left\| \mathbf{Q}^{(n)} \cdots \mathbf{Q}^{(1)} \right\|_{\infty}^{\frac{1}{n}} \mid \mathbf{Q}^{(i)} \in \mathcal{Q} \text{ for all } i \in [n] \right\} .$$

However, for $\{\mathrm{C1+}\}$, consider

$$\mathbf{Q} = \begin{bmatrix} 0.5 & 0.7 \\ 0 & 0.7 \end{bmatrix}$$

with spectral radius $\rho(\mathbf{Q}) = 0.7 < 1$, yet we have
$\widetilde{\gamma}^2 = \left\| \mathbf{Q}^2 \mathbf{1} \right\|_{\infty} = 1.49 > 1$.

Lemma 10 (New analysis)

For Equation (1) *under Condition* C1+, *initialized at any point,* VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}) .$$

Comparison to the classic analysis for halting SSGs

Lemma 11

$$(1 - o(1))\,(1 - \widetilde{\gamma}) \le (1 - \gamma)\,.$$

Example 1

Consider $\mathcal{Q} = \{\mathbf{Q}\}$ where

$$\mathbf{Q} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & \dots & 0 \\ 0 & 1/2 & 1/2 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & 1/2 & 1/2 \\ 0 & \dots & \dots & \dots & 0 & 1/2 \end{pmatrix} \in \mathbb{R}^{n \times n},$$

we have $1 - \widetilde{\gamma} \le \frac{1}{n\cdot(2^n - 1)} \ll (1 - o(1)) \cdot \frac{1}{2} = 1 - \gamma$.

Comparison to the classic analysis for halting SSGs

Lemma 11

$$(1 - o(1))\,(1 - \widetilde{\gamma}) \le (1 - \gamma)\,.$$

Example 1

Consider $\mathcal{Q} = \{\mathbf{Q}\}$ where

$$\mathbf{Q} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & \dots & 0 \\ 0 & 1/2 & 1/2 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & 1/2 & 1/2 \\ 0 & \dots & \dots & \dots & 0 & 1/2 \end{pmatrix} \in \mathbb{R}^{n \times n}\,,$$

we have $1 - \widetilde{\gamma} \le \frac{1}{n \cdot (2^n - 1)} \ll (1 - o(1)) \cdot \frac{1}{2} = 1 - \gamma$.

Summary: Algorithms for absolutely halting subclass

Theorem 3

For Equation (1) *under Condition* C1+*:*

- PI *diverges.*
- VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}).$$

Remark

Restricted to halting SSGs:

- Our new analysis of VI is comparable to the classic analysis, and
- perhaps surprisingly, can even be exponentially faster in some cases.

Summary: Algorithms for absolutely halting subclass

Theorem 3

For Equation (1) *under Condition* C1+*:*

- PI *diverges.*
- VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}).$$

Remark

Restricted to halting SSGs:

- Our new analysis of VI is comparable to the classic analysis, and
- perhaps surprisingly, can even be exponentially faster in some cases.

Summary: Algorithms for absolutely halting subclass

Theorem 3

For Equation (1) *under Condition* C1+*:*

- PI *diverges.*
- VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}).$$

Remark

Restricted to halting SSGs:

- Our new analysis of VI is comparable to the classic analysis, and
- perhaps surprisingly, can even be exponentially faster in some cases.

Summary: Algorithms for absolutely halting subclass

Theorem 3

For Equation (1) *under Condition* C1+*:*

- PI *diverges.*
- VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}).$$

Remark

Restricted to halting SSGs:

- Our new analysis of VI is comparable to the classic analysis, and
- perhaps surprisingly, can even be exponentially faster in some cases.

Summary: Algorithms for absolutely halting subclass

Theorem 3

For Equation (1) *under Condition* C1+*:*

- PI *diverges.*
- VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}).$$

Remark

Restricted to halting SSGs:

- Our new analysis of VI is comparable to the classic analysis, and
- perhaps surprisingly, can even be exponentially faster in some cases.

Summary: Algorithms for absolutely halting subclass

Theorem 3

For Equation (1) *under Condition* C1+*:*

- PI *diverges.*
- VI *converges linearly to the solution at a ratio of*

$$\gamma = (1 + o(1)) \cdot \max_{\mathbf{Q} \in \mathcal{Q}} \rho(\mathbf{Q}).$$

Remark

Restricted to halting SSGs:

- Our new analysis of VI is comparable to the classic analysis, and
- perhaps surprisingly, can even be exponentially faster in some cases.

Outline

1. Introduction
   - Stochastic games
   - Optimization problem
2. Motivating examples and restrictive conditions
   - Motivating examples
   - Restrictive conditions
3. Complexity
   - Complexity of the decision problem
   - Complexity of checking the conditions
4. Algorithms
   - Classic algorithms for halting SSGs
   - Algorithms for absolutely halting subclass
   - Algorithms for halting subclass

Value iteration under Condition C1

A counterexample satisfying Conditions C1, C3 and C4:

$$\begin{cases} x_1 = \max\{x_3, x_4\}, \\ x_2 = \max\{x_5, x_6\}, \\ x_3 = -0.9x_1 + 1.8x_2 - 1.5, \\ x_4 = 0.5x_1 + 1.5x_2 - 1.5, \\ x_5 = -0.5x_1 - 1.0, \\ x_6 = -0.25x_1 - 0.25x_2 - 1.0. \end{cases}$$

Initialized at

$$\mathbf{x}^{(0)} = \left[{}^{3}/_{7}, {}^{5}/_{7}, -{}^{3}/_{5}, -{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{9}/_{7}\right]^{\mathsf{T}},$$

VI gets

$$\mathbf{x}^{(1)} = \left[-{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{3}/_{5}, -{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{9}/_{7}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(2)} = \left[-{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{489}/_{140}, -{}^{24}/_{7}, -{}^{21}/_{28}, -{}^{9}/_{14}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(3)} = \left[-{}^{24}/_{7}, -{}^{9}/_{14}, -{}^{489}/_{140}, -{}^{24}/_{7}, -{}^{21}/_{28}, -{}^{9}/_{14}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(4)} = \left[-{}^{24}/_{7}, -{}^{9}/_{14}, {}^{3}/_{7}, -{}^{117}/_{28}, {}^{5}/_{7}, {}^{1}/_{56}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(5)} = \left[{}^{3}/_{7}, {}^{5}/_{7}, {}^{3}/_{7}, -{}^{117}/_{28}, {}^{5}/_{7}, {}^{1}/_{56}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(6)} = \left[{}^{3}/_{7}, {}^{5}/_{7}, -{}^{3}/_{5}, -{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{9}/_{7}\right]^{\mathsf{T}},$$

$\cdots\cdots$

and enters a loop.

Value iteration under Condition C1

Initialized at

$$\mathbf{x}^{(0)} = \left[{}^{3}/_{7}, {}^{5}/_{7}, -{}^{3}/_{5}, -{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{9}/_{7}\right]^{\mathsf{T}},$$

A counterexample satisfying Conditions C1, C3 and C4:

$$\begin{cases} x_1 = \max\{x_3, x_4\}, \\ x_2 = \max\{x_5, x_6\}, \\ x_3 = -0.9x_1 + 1.8x_2 - 1.5, \\ x_4 = 0.5x_1 + 1.5x_2 - 1.5, \\ x_5 = -0.5x_1 - 1.0, \\ x_6 = -0.25x_1 - 0.25x_2 - 1.0. \end{cases}$$

VI gets

$$\mathbf{x}^{(1)} = \left[-{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{3}/_{5}, -{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{9}/_{7}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(2)} = \left[-{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{489}/_{140}, -{}^{24}/_{7}, -{}^{21}/_{28}, -{}^{9}/_{14}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(3)} = \left[-{}^{24}/_{7}, -{}^{9}/_{14}, -{}^{489}/_{140}, -{}^{24}/_{7}, -{}^{21}/_{28}, -{}^{9}/_{14}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(4)} = \left[-{}^{24}/_{7}, -{}^{9}/_{14}, {}^{3}/_{7}, -{}^{117}/_{28}, {}^{5}/_{7}, {}^{1}/_{56}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(5)} = \left[{}^{3}/_{7}, {}^{5}/_{7}, {}^{3}/_{7}, -{}^{117}/_{28}, {}^{5}/_{7}, {}^{1}/_{56}\right]^{\mathsf{T}},$$
$$\mathbf{x}^{(6)} = \left[{}^{3}/_{7}, {}^{5}/_{7}, -{}^{3}/_{5}, -{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{9}/_{7}\right]^{\mathsf{T}},$$

$\cdots\cdots$

and enters a loop.

Value iteration under Condition C1

A counterexample satisfying Conditions C1, C3 and C4:

$$\begin{cases} x_1 = \max\{x_3, x_4\}, \\ x_2 = \max\{x_5, x_6\}, \\ x_3 = -0.9x_1 + 1.8x_2 - 1.5, \\ x_4 = 0.5x_1 + 1.5x_2 - 1.5, \\ x_5 = -0.5x_1 - 1.0, \\ x_6 = -0.25x_1 - 0.25x_2 - 1.0. \end{cases}$$

Initialized at

$$\mathbf{x}^{(0)} = [{}^{3}/_{7}, {}^{5}/_{7}, -{}^{3}/_{5}, -{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{9}/_{7}]^{\mathsf{T}},$$

VI gets

$$\mathbf{x}^{(1)} = [-{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{3}/_{5}, -{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{9}/_{7}]^{\mathsf{T}},$$
$$\mathbf{x}^{(2)} = [-{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{489}/_{140}, -{}^{24}/_{7}, -{}^{21}/_{28}, -{}^{9}/_{14}]^{\mathsf{T}},$$
$$\mathbf{x}^{(3)} = [-{}^{24}/_{7}, -{}^{9}/_{14}, -{}^{489}/_{140}, -{}^{24}/_{7}, -{}^{21}/_{28}, -{}^{9}/_{14}]^{\mathsf{T}},$$
$$\mathbf{x}^{(4)} = [-{}^{24}/_{7}, -{}^{9}/_{14}, {}^{3}/_{7}, -{}^{117}/_{28}, {}^{5}/_{7}, {}^{1}/_{56}]^{\mathsf{T}},$$
$$\mathbf{x}^{(5)} = [{}^{3}/_{7}, {}^{5}/_{7}, {}^{3}/_{7}, -{}^{117}/_{28}, {}^{5}/_{7}, {}^{1}/_{56}]^{\mathsf{T}},$$
$$\mathbf{x}^{(6)} = [{}^{3}/_{7}, {}^{5}/_{7}, -{}^{3}/_{5}, -{}^{3}/_{14}, -{}^{17}/_{14}, -{}^{9}/_{7}]^{\mathsf{T}},$$
$$\cdots\cdots$$

and enters a loop.

Simple policy iteration

Algorithm 3 Simple Policy Iteration (SPI)

1: Let $\mathcal{I}(i)$ be a permutation of $\mathcal{N}(i)$, $i \in [1, n_1 + n_2]$.
2: Let $\mathbf{Q} \in \mathcal{Q}$ such that

$$\mathbf{Q}_{i,\cdot} = \begin{cases} \mathbf{e}_{\mathcal{I}(i)(1)}, & i \in [1, n_1 + n_2], \\ \mathbf{q}_i, & i \in (n_1 + n_2, n]. \end{cases}$$

3: Let $\boldsymbol{\ell} \leftarrow (1, \cdots, 1)$.
4: **loop**
5: $\quad \mathbf{x} \leftarrow (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{b}$
6: $\quad \Gamma \leftarrow \{i \in [1, n_1] \mid \exists k \in \mathcal{N}(i)$ such that $x_k < x_i\} \cup \{j \in (n_1, n_1 + n_2] \mid \exists l \in \mathcal{N}(j)$ such that $x_l > x_j\}$
7: $\quad$ **if** Γ is empty **then**
8: $\quad\quad$ **return x**
9: $\quad$ **else**
10: $\quad\quad k \leftarrow \min \ \Gamma$
11: $\quad\quad \ell_k \leftarrow \begin{cases} 1, \text{ if } \ell_k = |\mathcal{N}(k)|, \\ \ell_k + 1, \text{ otherwise.} \end{cases}$
12: $\quad\quad \mathbf{Q}_{k,\cdot} \leftarrow \mathbf{e}^{\mathsf{T}}_{\mathcal{I}(k)(\ell_k)}$
13: $\quad$ **end if**
14: **end loop**

Randomized simple policy iteration

Algorithm 4 Randomized Simple Policy Iteration (RandSPI)

if the number of min and max variables $k > 0$ **then**
  Generate a new uniform random permutation $\mathcal{I}(k)$ of $\mathcal{N}(k)$
  for $i = 1, \cdots, |\mathcal{N}(k)|$ **do**
    Replace the kth equation by $x_k = x_{\mathcal{I}(k)(i)}$
    Recursively apply the algorithm to the new system of equations to find a solution $\mathbf{x}$
    if $\mathbf{x}$ satisfies the original kth equation **then**
      return x
    end if
  end for
else
  Let $\mathcal{Q} = \{\mathbf{Q}\}$
  return $(\mathbf{I} - \mathbf{Q})^{-1}\mathbf{b}$
end if

Summary: Algorithms for halting subclass

Theorem 4

For Equation (1) *under Condition* C1*:*

- *Both* PI *and* VI *diverge.*
- SPI *returns the exact solution in no more than* $\Pi_{i\in[n_1+n_2]}|\mathcal{N}(i)|$ *iterations.*
- RandSPI *returns the exact solution in no more than*

$$\frac{\Pi_{i\in[n_1+n_2]}(|\mathcal{N}(i)|+1)}{2^{n_1+n_2}}$$

recursive calls in expectation.

Remark

SPI and RandSPI are cleverer than brute force!

Summary: Algorithms for halting subclass

Theorem 4

For Equation (1) *under Condition* C1*:*

- *Both* PI *and* VI *diverge.*
- SPI *returns the exact solution in no more than* $\mathit{\Pi}_{i\in[n_1+n_2]} |\mathcal{N}(i)|$ *iterations.*
- RandSPI *returns the exact solution in no more than*

$$\frac{\mathit{\Pi}_{i\in[n_1+n_2]} (|\mathcal{N}(i)| + 1)}{2^{n_1+n_2}}$$

recursive calls in expectation.

Remark

SPI and RandSPI are cleverer than brute force!

Summary: Algorithms for halting subclass

Theorem 4

For Equation (1) *under Condition* C1*:*

- *Both* PI *and* VI *diverge.*
- SPI *returns the exact solution in no more than* $\mathit{\Pi}_{i \in [n_1+n_2]} |\mathcal{N}(i)|$ *iterations.*
- RandSPI *returns the exact solution in no more than*

$$\frac{\mathit{\Pi}_{i \in [n_1+n_2]} (|\mathcal{N}(i)| + 1)}{2^{n_1+n_2}}$$

recursive calls in expectation.

Remark

SPI and RandSPI are cleverer than brute force!

Summary: Algorithms for halting subclass

Theorem 4

For Equation (1) *under Condition* C1*:*

- *Both* PI *and* VI *diverge.*
- SPI *returns the exact solution in no more than* $\Pi_{i\in[n_1+n_2]} |\mathcal{N}(i)|$ *iterations.*
- RandSPI *returns the exact solution in no more than*

$$\frac{\Pi_{i\in[n_1+n_2]} (|\mathcal{N}(i)| + 1)}{2^{n_1+n_2}}$$

recursive calls in expectation.

Remark

SPI and RandSPI are cleverer than brute force!

Summary: Algorithms for halting subclass

Theorem 4

For Equation (1) *under Condition* C1*:*

- *Both* PI *and* VI *diverge.*
- SPI *returns the exact solution in no more than* $\Pi_{i \in [n_1+n_2]} |\mathcal{N}(i)|$ *iterations.*
- RandSPI *returns the exact solution in no more than*

$$\frac{\Pi_{i \in [n_1+n_2]} (|\mathcal{N}(i)| + 1)}{2^{n_1+n_2}}$$

recursive calls in expectation.

Remark

SPI and RandSPI are cleverer than brute force!

Conclusion

Subclasses	Decision	Checking	PI	VI	SPI
{C1}	UP ∩ coUP (SSG-hard)	coNP-*comp.*	✘	✘	✔
{C1+}		PTIME	✘	✔	✔
{C1, C2, C3}		PTIME	✓	✓	✓

Table 9: The complexity and algorithms of problem subclasses.