Reducing Stochastic Games to Semidefinite Programming

Manuel Bodirsky

Institut für Algebra, TU Dresden

Joint work with Georg Logo and Mateusz Skomra

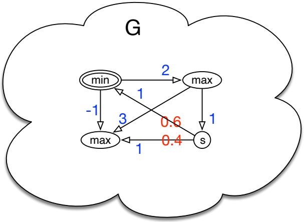
Jena, 3.3.2025

European Research Council
Established by the European Commission

ERC Synergy Grant POCOCOP (GA 101071674).

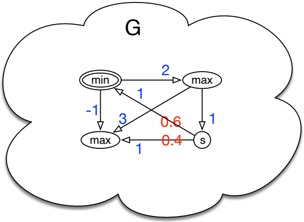
Outline

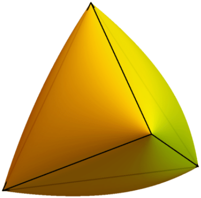
1. (Stochastic) Mean Payoff Games



Outline

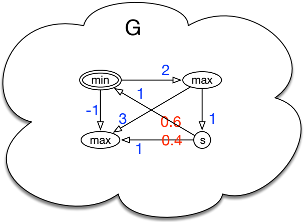
1. (Stochastic) Mean Payoff Games
2. Semidefinite Programming (SDP)



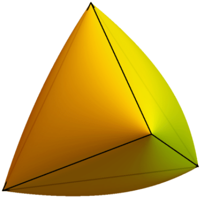


Outline

1. (Stochastic) Mean Payoff Games



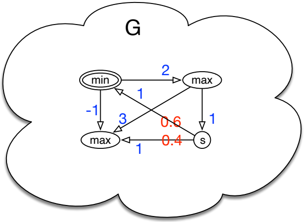
2. Semidefinite Programming (SDP)



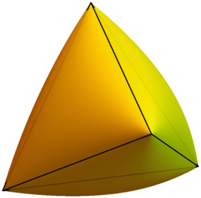
3. From Stochastic Games to Max-Average Constraints

Outline

1. (Stochastic) Mean Payoff Games



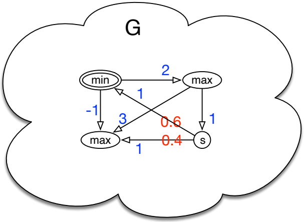
2. Semidefinite Programming (SDP)



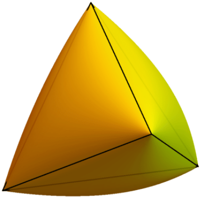
3. From Stochastic Games to Max-Average Constraints
4. From Max-Average Constraints to non-archimedian SDPs

Outline

1. (Stochastic) Mean Payoff Games



2. Semidefinite Programming (SDP)

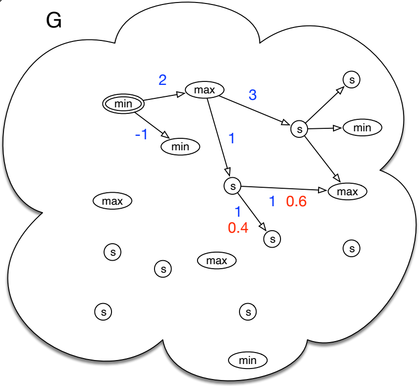


3. From Stochastic Games to Max-Average Constraints
4. From Max-Average Constraints to non-archimedian SDPs
5. From non-archimdedian SDPs to real SDPs

(Stochastic) Mean Payoff Games

(Stochastic) Mean Payoff Games

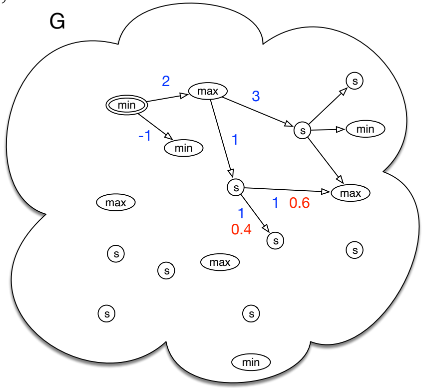
$$\text{Game } G = (\underbrace{V_{\text{min}} \cup V_{\text{max}} \cup V_{\text{rand}}}_{=:V}, E, r, s, p)$$



(Stochastic) Mean Payoff Games

Game $G = (\underbrace{V_{\min} \cup V_{\max} \cup V_{\text{rand}}}_{=:V}, E, r, s, p)$

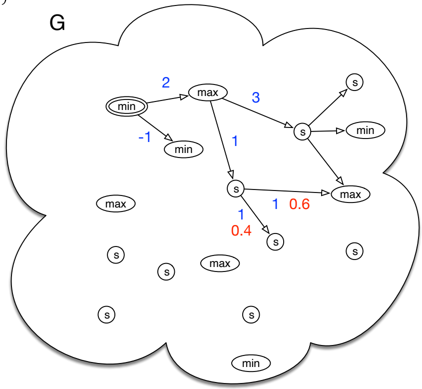
- (Pure positional) strategies:
  $\sigma\colon V_{\min} \to V$ and $\tau\colon V_{\max} \to V$



(Stochastic) Mean Payoff Games

Game $G = (\underbrace{V_{\min} \cup V_{\max} \cup V_{\text{rand}}}_{=:V}, E, r, s, p)$

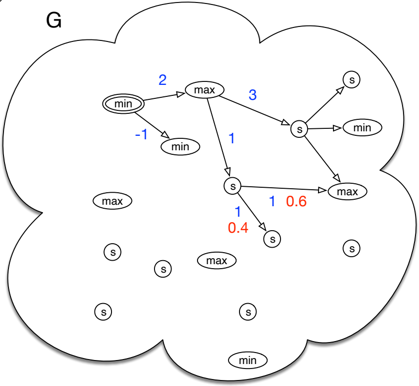
- (Pure positional) strategies:
  $\sigma\colon V_{\min} \to V$ and $\tau\colon V_{\max} \to V$
- Only stochastic vertices:
  Markov Chain.



(Stochastic) Mean Payoff Games

Game $G = (\underbrace{V_{\min} \cup V_{\max} \cup V_{\text{rand}}}_{=:V}, E, r, s, p)$

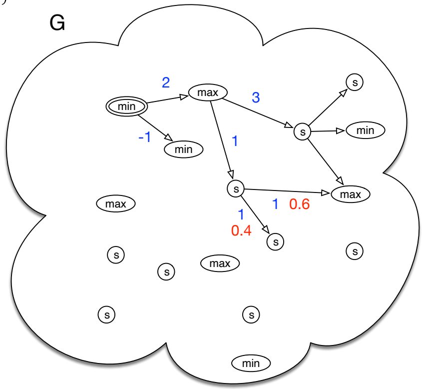
- (Pure positional) strategies:
  $\sigma\colon V_{\min} \to V$ and $\tau\colon V_{\max} \to V$
- Only stochastic vertices:
  Markov Chain.
- Mean payoff reward:
  $\lim_{n\to\infty} \frac{1}{n} E_{\sigma,\tau}(r_{su_1} + \cdots + r_{u_{n-1}u_n})$.



(Stochastic) Mean Payoff Games

Game $G = (\underbrace{V_{min} \cup V_{max} \cup V_{rand}}_{=:V}, E, r, s, p)$

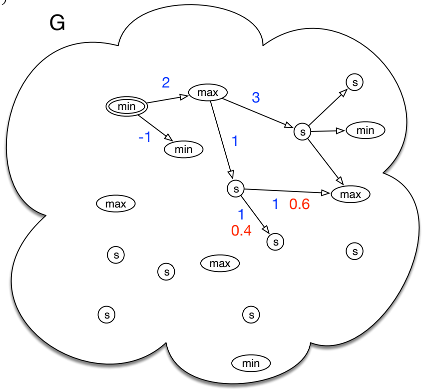
- (Pure positional) strategies:
  $\sigma \colon V_{min} \to V$ and $\tau \colon V_{max} \to V$
- Only stochastic vertices:
  Markov Chain.
- Mean payoff reward:
  $\lim_{n\to\infty} \frac{1}{n} E_{\sigma,\tau}(r_{su_1} + \cdots + r_{u_{n-1}u_n})$.
- Fact (Liggett+Lippman'1969):
  there exist strategies σ^*, τ^* s.t.
  $g(\sigma^*, \tau) \leq \underbrace{g(\sigma^*, \tau^*)}_{=:\textit{value} \text{ of } G} \leq g(\sigma, \tau^*)$
  for all strategies σ, τ.



(Stochastic) Mean Payoff Games

Game $G = (\underbrace{V_{\min} \cup V_{\max} \cup V_{\text{rand}}}_{=:V}, E, r, s, p)$

- (Pure positional) strategies:
  $\sigma\colon V_{\min} \to V$ and $\tau\colon V_{\max} \to V$
- Only stochastic vertices:
  Markov Chain.
- Mean payoff reward:
  $\lim_{n\to\infty} \frac{1}{n} E_{\sigma,\tau}(r_{su_1} + \cdots + r_{u_{n-1}u_n})$.
- Fact (Liggett+Lippman'1969):
  there exist strategies σ^*, τ^* s.t.
  $g(\sigma^*, \tau) \leq \underbrace{g(\sigma^*, \tau^*)}_{=:\textit{value} \text{ of } G} \leq g(\sigma, \tau^*)$
  for all strategies σ, τ.
- No stochastic vertices:
  Mean Payoff Game (MPG).



Complexity of (Stochastic) MPGs

Complexity of (Stochastic) MPGs

- Computational problem: given G, compute value of G.

Complexity of (Stochastic) MPGs

- Computational problem: given G, compute value of G.
- Special case simple stochastic games:
  stochastic nodes have 2 outgoing edges, each with probability $1/2$.
  Special terminal nodes W and L with loops, with $r_{WW} = 1$ and $r_{LL} = 0$.

Complexity of (Stochastic) MPGs

- Computational problem: given G, compute value of G.
- Special case simple stochastic games:
  stochastic nodes have 2 outgoing edges, each with probability $1/2$.
  Special terminal nodes W and L with loops, with $r_{WW} = 1$ and $r_{LL} = 0$.
- A simple stochastic game is called stopping if for any pair of strategies, we reach W or L with probability one.

Complexity of (Stochastic) MPGs

- Computational problem: given G, compute value of G.
- Special case simple stochastic games:
  stochastic nodes have 2 outgoing edges, each with probability $1/2$.
  Special terminal nodes W and L with loops, with $r_{WW} = 1$ and $r_{LL} = 0$.
- A simple stochastic game is called stopping if for any pair of strategies, we reach W or L with probability one.
- There is a polynomial-time Turing-reduction from computing the value of a stochastic mean payoff game to deciding whether the value of a stopping simple stochastic game is at least $1/2$ (Allamigeon+Gaubert+Skomra'2018).

Complexity of (Stochastic) MPGs

- Computational problem: given G, compute value of G.
- Special case simple stochastic games:
  stochastic nodes have 2 outgoing edges, each with probability $1/2$.
  Special terminal nodes W and L with loops, with $r_{WW} = 1$ and $r_{LL} = 0$.
- A simple stochastic game is called stopping if for any pair of strategies, we reach W or L with probability one.
- There is a polynomial-time Turing-reduction from computing the value of a stochastic mean payoff game to deciding whether the value of a stopping simple stochastic game is at least $1/2$ (Allamigeon+Gaubert+Skomra'2018).
- Problem is in NP $\cap$ coNP (Condon'1992).

Complexity of (Stochastic) MPGs

- Computational problem: given G, compute value of G.
- Special case simple stochastic games:
  stochastic nodes have 2 outgoing edges, each with probability $1/2$.
  Special terminal nodes W and L with loops, with $r_{WW} = 1$ and $r_{LL} = 0$.
- A simple stochastic game is called stopping if for any pair of strategies, we reach W or L with probability one.
- There is a polynomial-time Turing-reduction from computing the value of a stochastic mean payoff game to deciding whether the value of a stopping simple stochastic game is at least $1/2$ (Allamigeon+Gaubert+Skomra'2018).
- Problem is in NP $\cap$ coNP (Condon'1992).
- Not known to be in P, already for (deterministic) MPGs.

Complexity of (Stochastic) MPGs

- Computational problem: given G, compute value of G.
- Special case simple stochastic games:
  stochastic nodes have 2 outgoing edges, each with probability $1/2$.
  Special terminal nodes W and L with loops, with $r_{WW} = 1$ and $r_{LL} = 0$.
- A simple stochastic game is called stopping if for any pair of strategies, we reach W or L with probability one.
- There is a polynomial-time Turing-reduction from computing the value of a stochastic mean payoff game to deciding whether the value of a stopping simple stochastic game is at least $1/2$ (Allamigeon+Gaubert+Skomra'2018).
- Problem is in NP $\cap$ coNP (Condon'1992).
- Not known to be in P, already for (deterministic) MPGs.
- Central open problem in verification.

Complexity of (Stochastic) MPGs

- Computational problem: given G, compute value of G.
- Special case simple stochastic games:
  stochastic nodes have 2 outgoing edges, each with probability $1/2$.
  Special terminal nodes W and L with loops, with $r_{WW} = 1$ and $r_{LL} = 0$.
- A simple stochastic game is called stopping if for any pair of strategies, we reach W or L with probability one.
- There is a polynomial-time Turing-reduction from computing the value of a stochastic mean payoff game to deciding whether the value of a stopping simple stochastic game is at least $1/2$ (Allamigeon+Gaubert+Skomra'2018).
- Problem is in NP $\cap$ coNP (Condon'1992).
- Not known to be in P, already for (deterministic) MPGs.
- Central open problem in verification.
- Parity games reduce to MPGs

Complexity of (Stochastic) MPGs

- Computational problem: given G, compute value of G.
- Special case simple stochastic games:
  stochastic nodes have 2 outgoing edges, each with probability $1/2$.
  Special terminal nodes W and L with loops, with $r_{WW} = 1$ and $r_{LL} = 0$.
- A simple stochastic game is called stopping if for any pair of strategies, we reach W or L with probability one.
- There is a polynomial-time Turing-reduction from computing the value of a stochastic mean payoff game to deciding whether the value of a stopping simple stochastic game is at least $1/2$ (Allamigeon+Gaubert+Skomra'2018).
- Problem is in NP $\cap$ coNP (Condon'1992).
- Not known to be in P, already for (deterministic) MPGs.
- Central open problem in verification.
- Parity games reduce to MPGs
- Quasi-polynomial algorithms for parity games (Calude+Jain+Khoussainov+Li+Stephan'2022) don't work for MPGs

Semidefinite Programming

Semidefinite Programming

- Recap: linear programming

$$\min_{x \in \mathbb{Q}^n} cx^\top \text{ subject to } Ax \leq b$$

Semidefinite Programming

- Recap: linear programming

$$\min_{x \in \mathbb{Q}^n} cx^\top \text{ subject to } Ax \leq b$$

known to be in P (Khachiyan'1979).

Semidefinite Programming

- Recap: linear programming

$$\min_{x \in \mathbb{Q}^n} cx^\top \text{ subject to } Ax \leq b$$

  known to be in P (Khachiyan'1979).
- Idea semidefinite programming: instead of $Ax \leq b$, allow more general class of constraints that are still
  - convex, and
  - semialgebraic.

Semidefinite Programming

- Recap: linear programming

$$\min_{x \in \mathbb{Q}^n} cx^\top \text{ subject to } Ax \leq b$$

  known to be in P (Khachiyan'1979).
- Idea semidefinite programming: instead of $Ax \leq b$, allow more general class of constraints that are still
  - convex, and
  - semialgebraic.
- $A \in \mathbb{R}^{k \times k}$ real symmetric matrix.

Semidefinite Programming

- Recap: linear programming

$$\min_{x \in \mathbb{Q}^n} cx^\top \text{ subject to } Ax \leq b$$

  known to be in P (Khachiyan'1979).
- Idea semidefinite programming: instead of $Ax \leq b$, allow more general class of constraints that are still
  - convex, and
  - semialgebraic.
- $A \in \mathbb{R}^{k \times k}$ real symmetric matrix.
  $A \succeq 0$: A is positive semidefinite, i.e., $y^\top Ay \geq 0$ for all $y \in \mathbb{R}^k$.

Semidefinite Programming

- Recap: linear programming

  $$\min_{x \in \mathbb{Q}^n} cx^\top \text{ subject to } Ax \leq b$$

  known to be in P (Khachiyan'1979).
- Idea semidefinite programming: instead of $Ax \leq b$,
  allow more general class of constraints that are still
  - convex, and
  - semialgebraic.
- $A \in \mathbb{R}^{k \times k}$ real symmetric matrix.
  $A \succeq 0$: A is positive semidefinite, i.e., $y^\top Ay \geq 0$ for all $y \in \mathbb{R}^k$.
- $S \subseteq \mathbb{R}^n$ is called a spectrahedron if there are symmetric matrices $A_0, A_1, \dots, A_n \in \mathbb{R}^{k \times k}$ such that

  $$S = \{(x_1, \dots, x_n) \mid \underbrace{A_0 + A_1 x_1 + \cdots + A_n x_n \succeq 0}_{\text{'linear matrix inequality (LMI)'}}\}$$

Semidefinite Programming

- Recap: linear programming

$$\min_{x \in \mathbb{Q}^n} cx^\top \text{ subject to } Ax \leq b$$

  known to be in P (Khachiyan'1979).
- Idea semidefinite programming: instead of $Ax \leq b$, allow more general class of constraints that are still
  - convex, and
  - semialgebraic.
- $A \in \mathbb{R}^{k \times k}$ real symmetric matrix.
  $A \succeq 0$: A is positive semidefinite, i.e., $y^\top Ay \geq 0$ for all $y \in \mathbb{R}^k$.
- $S \subseteq \mathbb{R}^n$ is called a spectrahedron if there are symmetric matrices $A_0, A_1, \ldots, A_n \in \mathbb{R}^{k \times k}$ such that

$$S = \{(x_1, \ldots, x_n) \mid \underbrace{A_0 + A_1 x_1 + \cdots + A_n x_n \succeq 0}_{\text{'linear matrix inequality (LMI)'}}\}$$

Example. $S = \left\{(x, y) \mid \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0\right\} = \left\{(x, y) \mid y - x^2 \geq 0\right\}$

Complexity of SDP

Complexity of SDP

- Many algorithmic approaches for LP can also be applied to SDP.

Complexity of SDP

- Many algorithmic approaches for LP can also be applied to SDP.
- Can often be solved efficiently in practise.

Complexity of SDP

- Many algorithmic approaches for LP can also be applied to SDP.
- Can often be solved efficiently in practise.
- But: the complexity of deciding the feasibility problem

$$\{(x_1, \dots, x_n) \mid A_0 + A_1 x_1 + \cdots + A_n x_n \succeq 0\} \stackrel{?}{=} \emptyset$$

for given symmetric $A_0, A_1, \dots, A_n \in \mathbb{Q}^{k \times k}$, is not known to be in P.

Complexity of SDP

- Many algorithmic approaches for LP can also be applied to SDP.
- Can often be solved efficiently in practise.
- But: the complexity of deciding the feasibility problem

$$\{(x_1, \dots, x_n) \mid A_0 + A_1 x_1 + \cdots + A_n x_n \succeq 0\} \stackrel{?}{=} \emptyset$$

  for given symmetric $A_0, A_1, \dots, A_n \in \mathbb{Q}^{k \times k}$, is not known to be in P.
- SDP $\in \exists\mathbb{R} \subseteq$ PSPACE.

Complexity of SDP

- Many algorithmic approaches for LP can also be applied to SDP.
- Can often be solved efficiently in practise.
- But: the complexity of deciding the feasibility problem

$$\{(x_1, \ldots, x_n) \mid A_0 + A_1 x_1 + \cdots + A_n x_n \succeq 0\} \overset{?}{=} \emptyset$$

  for given symmetric $A_0, A_1, \ldots, A_n \in \mathbb{Q}^{k \times k}$, is not known to be in P.
- SDP $\in \exists\mathbb{R} \subseteq$ PSPACE.
- As in LP, there are exact duals:
  if problem is in NP, then it is also in coNP (Ramana'1997)

Complexity of SDP

- Many algorithmic approaches for LP can also be applied to SDP.
- Can often be solved efficiently in practise.
- But: the complexity of deciding the feasibility problem

$$\{(x_1, \ldots, x_n) \mid A_0 + A_1 x_1 + \cdots + A_n x_n \succeq 0\} \stackrel{?}{=} \emptyset$$

  for given symmetric $A_0, A_1, \ldots, A_n \in \mathbb{Q}^{k \times k}$, is not known to be in P.
- SDP $\in \exists\mathbb{R} \subseteq$ PSPACE.
- As in LP, there are exact duals:
  if problem is in NP, then it is also in coNP (Ramana'1997)

Theorem (B.+Loho+Skomra'2025).

There is a polynomial-time reduction from simple stopping stochastic games to the feasibility problem for SDP.

Max-Average Constraints

An **instance** of the Max-average constraint satisfaction problem:
consists of conjunction of constraints of the form

- $x_0 \leq \max(x_1, \ldots, x_n)$
- $x_0 \leq \frac{x_1 + x_2}{2}$
- $x_0 = c$ for some constant $c \in \mathbb{Q}$.

Max-Average Constraints

An **instance** of the Max-average constraint satisfaction problem:
consists of conjunction of constraints of the form

- $x_0 \leq \max(x_1, \ldots, x_n)$
- $x_0 \leq \frac{x_1 + x_2}{2}$
- $x_0 = c$ for some constant $c \in \mathbb{Q}$.

Computational **task:** is there a solution over $\mathbb{Q} \cup \{-\infty\}$?

Max-Average Constraints

An **instance** of the Max-average constraint satisfaction problem:
consists of conjunction of constraints of the form

- $x_0 \leq \max(x_1, \ldots, x_n)$
- $x_0 \leq \frac{x_1 + x_2}{2}$
- $x_0 = c$ for some constant $c \in \mathbb{Q}$.

Computational **task:** is there a solution over $\mathbb{Q} \cup \{-\infty\}$?

Translation from stopping simple stochastic games to max-average CSPs:

Max-Average Constraints

An **instance** of the Max-average constraint satisfaction problem: consists of conjunction of constraints of the form

- $x_0 \leq \max(x_1, \ldots, x_n)$
- $x_0 \leq \frac{x_1 + x_2}{2}$
- $x_0 = c$ for some constant $c \in \mathbb{Q}$.

Computational **task:** is there a solution over $\mathbb{Q} \cup \{-\infty\}$?

Translation from stopping simple stochastic games to max-average CSPs:

$$
\begin{aligned}
x_i &\leq \max_{(i,j)\in E}\{x_j\} && \text{for every } i \in V_{\max}, \\
x_i &\leq \min_{(i,j)\in E}\{x_j\} && \text{for every } i \in V_{\min}, \\
x_i &\leq \frac{1}{2} \sum_{(i,j)\in E} x_j && \text{for every } i \in V_{\text{rand}}, \\
x_i &\geq 0 && \text{for every } i \in V \\
x_L &= 0, x_W = 1, x_s \geq 1/2,
\end{aligned}
$$

Max-Average Constraints

An **instance** of the Max-average constraint satisfaction problem: consists of conjunction of constraints of the form

- $x_0 \leq \max(x_1, \ldots, x_n)$
- $x_0 \leq \frac{x_1 + x_2}{2}$
- $x_0 = c$ for some constant $c \in \mathbb{Q}$.

Computational **task:** is there a solution over $\mathbb{Q} \cup \{-\infty\}$?

Translation from stopping simple stochastic games to max-average CSPs:

$$\begin{aligned}
x_i &\leq \max_{(i,j) \in E}\{x_j\} && \text{for every } i \in V_{\max}, \\
x_i &\leq \min_{(i,j) \in E}\{x_j\} && \text{for every } i \in V_{\min}, \\
x_i &\leq \frac{1}{2} \sum_{(i,j) \in E} x_j && \text{for every } i \in V_{\text{rand}}, \\
x_i &\geq 0 && \text{for every } i \in V \\
x_L &= 0, x_W = 1, x_s \geq 1/2,
\end{aligned}$$

See, e.g., Bertrand+Bouyer-Decitre+Fijalkov+Skomra'2023.

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

$$\mathtt{x} = c_0 t^{a/n} + c_1 t^{(a-1)/n} + c_2 t^{(a-2)/n} + \dots$$

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

$$\mathtt{x} = c_0 t^{a/n} + c_1 t^{(a-1)/n} + c_2 t^{(a-2)/n} + \dots$$

a real-closed field.

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

$$\mathtt{x} = c_0 t^{a/n} + c_1 t^{(a-1)/n} + c_2 t^{(a-2)/n} + \dots$$

a real-closed field.

$$\mathtt{x}_{i_0} \leq \mathtt{x}_{i_1} + \cdots + \mathtt{x}_{i_k} \quad \text{for every constraint } x_{i_0} \leq \max(x_{i_1}, \dots, x_{i_k}) \text{ in } \phi,$$

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

$$\mathtt{x} = c_0 t^{a/n} + c_1 t^{(a-1)/n} + c_2 t^{(a-2)/n} + \dots$$

a real-closed field.

$$\mathtt{x}_{i_0} \leq \mathtt{x}_{i_1} + \cdots + \mathtt{x}_{i_k} \quad \text{for every constraint } x_{i_0} \leq \max(x_{i_1}, \dots, x_{i_k}) \text{ in } \phi,$$

$$\mathtt{x}_{i_0}^2 \leq \mathtt{x}_{i_1}\mathtt{x}_{i_2} \quad \text{for every constraint } x_{i_0} \leq \frac{x_{i_1} + x_{i_2}}{2} \text{ in } \phi,$$

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

$$\mathtt{x} = c_0 t^{a/n} + c_1 t^{(a-1)/n} + c_2 t^{(a-2)/n} + \dots$$

a real-closed field.

$$\mathtt{x}_{i_0} \leq \mathtt{x}_{i_1} + \cdots + \mathtt{x}_{i_k} \quad \text{for every constraint } x_{i_0} \leq \max(x_{i_1}, \dots, x_{i_k}) \text{ in } \phi,$$

$$\mathtt{x}_{i_0}^2 \leq \mathtt{x}_{i_1} \mathtt{x}_{i_2} \quad \text{for every constraint } x_{i_0} \leq \frac{x_{i_1} + x_{i_2}}{2} \text{ in } \phi,$$

$$\mathtt{x}_{i_0} = t^c \quad \text{for every constraint } x_{i_0} = c \text{ in } \phi,$$

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

$$\mathtt{x} = c_0 t^{a/n} + c_1 t^{(a-1)/n} + c_2 t^{(a-2)/n} + \dots$$

a real-closed field.

$\mathtt{x}_{i_0} \leq \mathtt{x}_{i_1} + \cdots + \mathtt{x}_{i_k}$	for every constraint $x_{i_0} \leq \max(x_{i_1}, \dots, x_{i_k})$ in ϕ,
$\mathtt{x}_{i_0}^2 \leq \mathtt{x}_{i_1} \mathtt{x}_{i_2}$	for every constraint $x_{i_0} \leq \frac{x_{i_1} + x_{i_2}}{2}$ in ϕ,
$\mathtt{x}_{i_0} = t^c$	for every constraint $x_{i_0} = c$ in ϕ,
$\mathtt{x}_{i_0} \geq 0$	for every $i_0 \in \{1, \dots, n\}$.

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

$$\mathtt{x} = c_0 t^{a/n} + c_1 t^{(a-1)/n} + c_2 t^{(a-2)/n} + \dots$$

a real-closed field.

$$\begin{array}{ll} \mathtt{x}_{i_0} \leq \mathtt{x}_{i_1} + \cdots + \mathtt{x}_{i_k} & \text{for every constraint } x_{i_0} \leq \max(x_{i_1}, \dots, x_{i_k}) \text{ in } \phi, \\ \mathtt{x}_{i_0}^2 \leq \mathtt{x}_{i_1} \mathtt{x}_{i_2} & \text{for every constraint } x_{i_0} \leq \dfrac{x_{i_1} + x_{i_2}}{2} \text{ in } \phi, \\ \mathtt{x}_{i_0} = t^c & \text{for every constraint } x_{i_0} = c \text{ in } \phi, \\ \mathtt{x}_{i_0} \geq 0 & \text{for every } i_0 \in \{1, \dots, n\}. \end{array}$$

Max-average instance satisfiable if and only if this SDP has solution in K.

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

$$\mathtt{x} = c_0 t^{a/n} + c_1 t^{(a-1)/n} + c_2 t^{(a-2)/n} + \dots$$

a real-closed field.

$\mathtt{x}_{i_0} \leq \mathtt{x}_{i_1} + \cdots + \mathtt{x}_{i_k}$	for every constraint $x_{i_0} \leq \max(x_{i_1}, \dots, x_{i_k})$ in ϕ,
$\mathtt{x}_{i_0}^2 \leq \mathtt{x}_{i_1}\mathtt{x}_{i_2}$	for every constraint $x_{i_0} \leq \frac{x_{i_1} + x_{i_2}}{2}$ in ϕ,
$\mathtt{x}_{i_0} = t^c$	for every constraint $x_{i_0} = c$ in ϕ,
$\mathtt{x}_{i_0} \geq 0$	for every $i_0 \in \{1, \dots, n\}$.

Max-average instance satisfiable if and only if this SDP has solution in *K*.

Idea:

- This reduction also works if t is replaced by a large number.

Non-archimedian SDPs

K: set of Puiseux series over $\mathbb{R}$

$$\mathrm{x} = c_0 t^{a/n} + c_1 t^{(a-1)/n} + c_2 t^{(a-2)/n} + \dots$$

a real-closed field.

$$\begin{array}{ll} \mathrm{x}_{i_0} \leq \mathrm{x}_{i_1} + \cdots + \mathrm{x}_{i_k} & \text{for every constraint } x_{i_0} \leq \max(x_{i_1}, \dots, x_{i_k}) \text{ in } \phi, \\ \mathrm{x}_{i_0}^2 \leq \mathrm{x}_{i_1} \mathrm{x}_{i_2} & \text{for every constraint } x_{i_0} \leq \frac{x_{i_1} + x_{i_2}}{2} \text{ in } \phi, \\ \mathrm{x}_{i_0} = t^c & \text{for every constraint } x_{i_0} = c \text{ in } \phi, \\ \mathrm{x}_{i_0} \geq 0 & \text{for every } i_0 \in \{1, \dots, n\}. \end{array}$$

Max-average instance satisfiable if and only if this SDP has solution in K.

Idea:

- This reduction also works if t is replaced by a large number.
- Using quantifier-elimination: double exponential is large enough.

Large Numbers

Problem: our reduction is not polynomial-time!

Large Numbers

Problem: our reduction is not polynomial-time!
(representation size of doubly exponential numbers!)

Large Numbers

Problem: our reduction is not polynomial-time!
(representation size of doubly exponential numbers!)

Easy to express $x \geq 2^{2^n}$ by SDP of polynomial size in n:

Large Numbers

Problem: our reduction is not polynomial-time!
(representation size of doubly exponential numbers!)

Easy to express $x \geq 2^{2^n}$ by SDP of polynomial size in n:

$$x_0 \geq 2,$$

Large Numbers

Problem: our reduction is not polynomial-time!
(representation size of doubly exponential numbers!)

Easy to express $x \geq 2^{2^n}$ by SDP of polynomial size in n:

$$x_0 \geq 2, x_1 \geq x_0^2,$$

Large Numbers

Problem: our reduction is not polynomial-time!
(representation size of doubly exponential numbers!)

Easy to express $x \geq 2^{2^n}$ by SDP of polynomial size in n:

$$x_0 \geq 2, x_1 \geq x_0^2, x_2 \geq x_1^2,$$

Large Numbers

Problem: our reduction is not polynomial-time!
(representation size of doubly exponential numbers!)

Easy to express $x \geq 2^{2^n}$ by SDP of polynomial size in n:

$$x_0 \geq 2, x_1 \geq x_0^2, x_2 \geq x_1^2, \ldots, x \geq x_{n-1}^2$$

Large Numbers

Problem: our reduction is not polynomial-time!
(representation size of doubly exponential numbers!)

Easy to express $x \geq 2^{2^n}$ by SDP of polynomial size in n:

$$x_0 \geq 2, x_1 \geq x_0^2, x_2 \geq x_1^2, \dots, x \geq x_{n-1}^2$$

Problem: need to express $x = 2^{2^n}$ for our reduction.

Large Numbers

Problem: our reduction is not polynomial-time!
(representation size of doubly exponential numbers!)

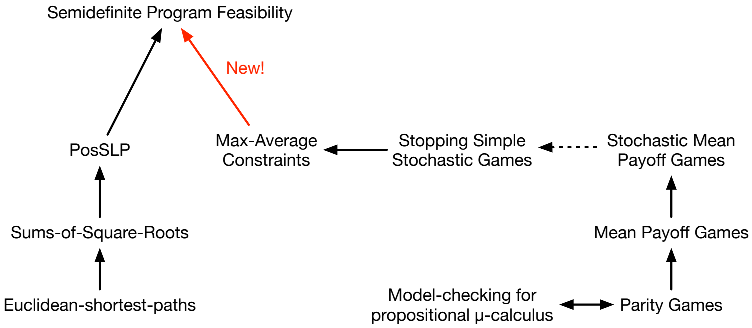
Easy to express $x \geq 2^{2^n}$ by SDP of polynomial size in n:

$$x_0 \geq 2, x_1 \geq x_0^2, x_2 \geq x_1^2, \ldots, x \geq x_{n-1}^2$$

Problem: need to express $x = 2^{2^n}$ for our reduction.

Solution: Use SDP duality theory to find small expression for $x = 2^{2^n}$.

Summary



Semidefinite Program Feasibility
New!
PosSLP
Max-Average Constraints
Stopping Simple Stochastic Games
Stochastic Mean Payoff Games
Sums-of-Square-Roots
Mean Payoff Games
Euclidean-shortest-paths
Model-checking for propositional μ-calculus
Parity Games