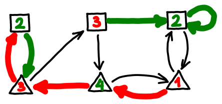
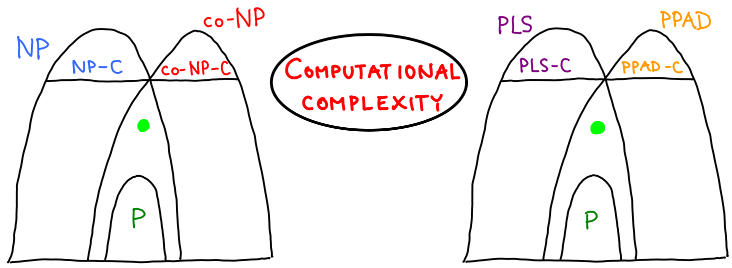
Algorithms for Solving Infinite Games on Graphs

Marcin Jurdziński

University of Warwick



How difficult is it to find a winning strategy?

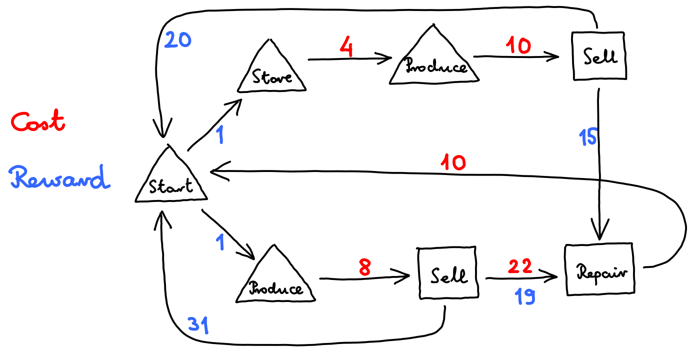


Plan

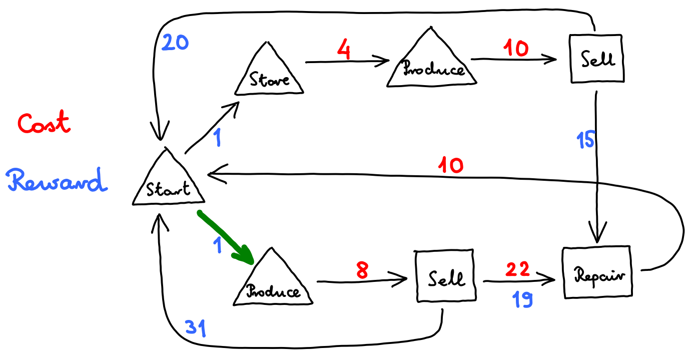
II Quantitative games

1. **Motivating examples**
2. Average-reward and discounted games
3. Dynamic programming: value iteration
4. Local search: strategy improvement
5. Mathematical programming: linear complementarity problems

An average cost-per-reward game

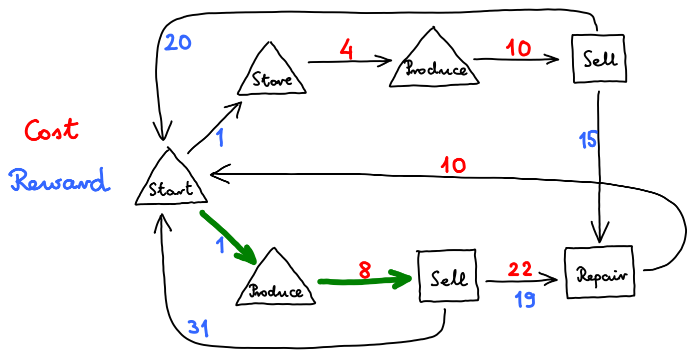


An average cost-per-reward game



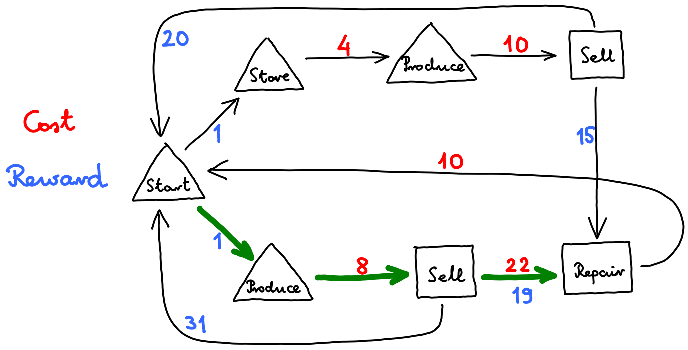
Average cost-per-reward: $\frac{0}{1}$

An average cost-per-reward game



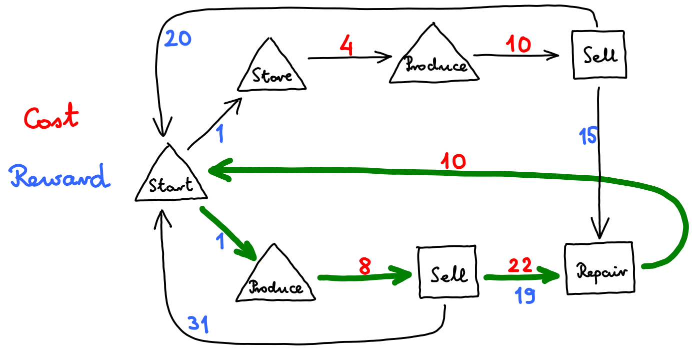
Average cost-per-reward: $\frac{0+8}{1+0} = 8$

An average cost-per-reward game



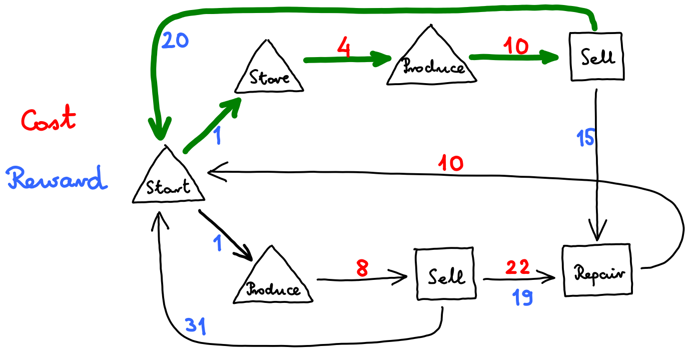
Average cost-per-reward: $\frac{0+8+22}{1+0+19} = \frac{3}{2}$

An average cost-per-reward game



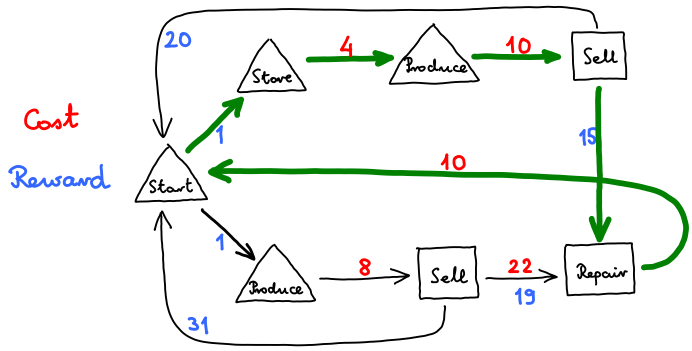
Average cost-per-reward: $\frac{0+8+22+10}{1+0+19+0} = 2$

An average cost-per-reward game



Average cost-per-reward: $\frac{0+4+10+0}{1+0+0+20} = \frac{2}{3}$

An average cost-per-reward game



Average cost-per-reward: $\frac{0+4+10+0+10}{1+0+0+15+0} = \frac{3}{2}$

Plan

II Quantitative games

1. Motivating examples
2. **Average-reward and discounted games**
3. Dynamic programming: value iteration
4. Local search: strategy improvement
5. Mathematical programming: linear complementarity problems

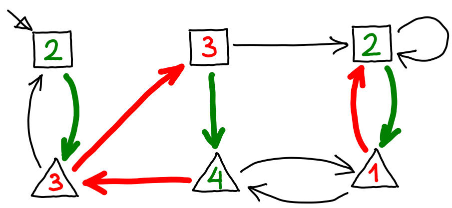
Parity Games

Players:

n vertices, m edges, d priorities

Even = □ = 0

Odd = △ = 1



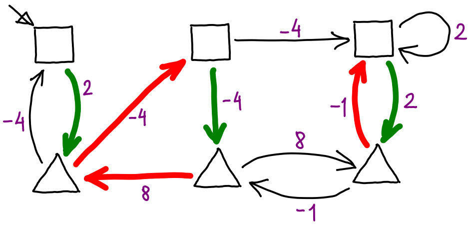
Winner of an infinite play:
parity of the highest priority occurring infinitely often

Average-reward games

Players:

Max = □

Min = △



Winner of an infinite play $\pi = \langle v_0, v_1, v_2, \dots \rangle$

Max if $\lim\limits_{n\to\infty} \left(\frac{1}{n} \cdot \sum_{i=0}^{n-1} r(v_i, v_{i+1}) \right) \geqslant 0$

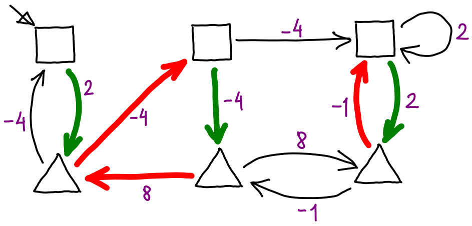
Discounted games

$0 < \delta < 1$

Players:

Max = □

Min = △



Winner of an infinite play $\pi = \langle v_0, v_1, v_2, \dots \rangle$

Max if $\sum_{i=0}^{\infty} \delta^i \cdot r(v_i, v_{i+1}) \geqslant 0$

Payoff, Value, Determinacy

$\pi = \langle s_0, s_1, s_2, \dots \rangle$

Average-reward: $A(\pi) = \lim\limits_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} r(s_i, s_{i+1})$

Discounted total: $D_\delta(\pi) = \sum_{i=0}^{\infty} \delta^i \cdot r(s_i, s_{i+1})$

Lower value: $\mathit{Val}_*(s) = \sup\limits_{\chi \in \Sigma_{Max}} \inf\limits_{\mu \in \Sigma_{Min}} P(\mathit{Play}(s, \mu, \chi))$

Upper value: $\mathit{Val}^*(s) = \inf\limits_{\mu \in \Sigma_{Min}} \sup\limits_{\chi \in \Sigma_{Max}} P(\mathit{Play}(s, \mu, \chi))$

Determinacy: $\mathit{Val}(s) = \mathit{Val}_*(s) = \mathit{Val}^*(s)$

Comparing computational complexity of infinite games

Thm [Puri 1995]
There is a polynomial-time reduction from parity games to average-reward games

Thm [1960's]
There is a polynomial-time reduction from average-reward games to discounted games

Thm [Hardy Littlewood 1930's]
If $\lim\limits_{n\to\infty} \left[\frac{1}{n} \cdot \sum_{i=0}^{n-1} a_i \right]$ exists
then $\lim\limits_{n\to\infty} \left[\frac{1}{n} \cdot \sum_{i=0}^{n-1} a_i \right] = \lim\limits_{\delta \nearrow 1} \left[(1-\delta) \cdot \sum_{i=0}^{\infty} \delta^i \cdot a_i \right]$

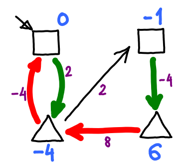
Plan

II Quantitative games

1. Motivating examples
2. Average-reward and discounted games
3. Dynamic programming: value iteration
4. Local search: strategy improvement
5. Mathematical programming: linear complementarity problems

Bellman equations for discounted games

$$\mathrm{Opt}(\Gamma): \quad \boxed{ v_s = \begin{cases} \max\limits_{(s,t)\in E} \left(r_{(s,t)} + \delta \cdot v_t \right) & \text{if } s \in S_{Max} \\ \min\limits_{(s,t)\in E} \left(r_{(s,t)} + \delta \cdot v_t \right) & \text{if } s \in S_{Min} \end{cases} }$$



$\delta = \frac{1}{2}$

Lemma If $V \models \mathrm{Opt}(\Gamma)$ then $V = \mathit{Val}^\Gamma$
and positional strategies choosing optimal successor are optimal

Bellman equations for discounted games

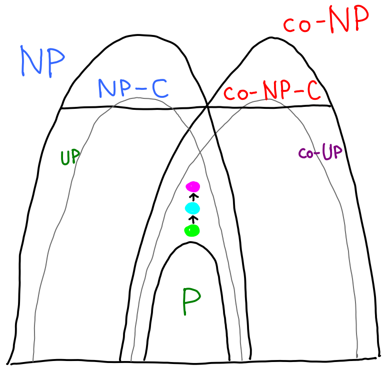
$$F : v_s \mapsto \begin{cases} \max\limits_{(s,t)\in E} \left(r_{(s,t)} + \delta \cdot v_t \right) & \text{if } s \in S_{Max} \\ \min\limits_{(s,t)\in E} \left(r_{(s,t)} + \delta \cdot v_t \right) & \text{if } s \in S_{Min} \end{cases}$$

Fact $F : \mathbb{R}^n \to \mathbb{R}^n$ is a contraction

Corollary

- F has a unique fixed point, i.e., $\mathrm{OPT}(\Gamma)$ has a solution
- Discounted, average reward, and parity games:
  - are positionally determined
  - are in $\mathrm{UP} \cap \text{co-}\mathrm{UP}$

Computational complexity of solving infinite games



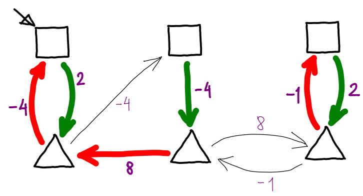
DISCOUNTED GAMES
AVERAGE-REWARD GAMES
PARITY GAMES

Plan

II Quantitative games

1. Motivating examples
2. Average-reward and discounted games
3. Dynamic programming: value iteration
4. **Local search: strategy improvement**
5. Mathematical programming: linear complementarity problems

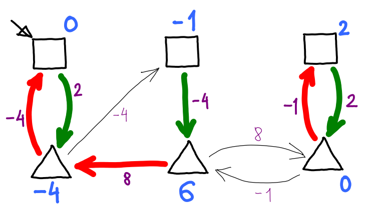
1-player Strategy Improvement



$\delta = \frac{1}{2}$

0. Pick $\mu \in \Pi_{Min}$
1. Find $V : S \to \mathbb{R}$, such that $V \models \mathrm{OPT}(\Gamma^{\mu})$
2. If $V \not\models \mathrm{OPT}(\Gamma)$
   then $\mu := \mathrm{Improve}(\mu, V)$; goto 1.

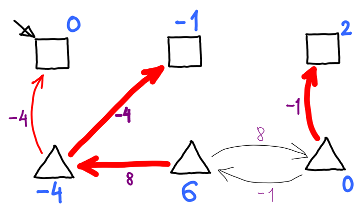
1-player Strategy Improvement



$\delta = \frac{1}{2}$

0. Pick $\mu \in \Pi_{Min}$
1. Find $V : S \to \mathbb{R}$, such that $V \models \mathrm{OPT}(\Gamma^{\mu})$
2. If $V \not\models \mathrm{OPT}(\Gamma)$
   then $\mu := \mathrm{Improve}(\mu, V)$; goto 1.

1-player Strategy Improvement



$\delta = \frac{1}{2}$

0. Pick $\mu \in \Pi_{Min}$
1. Find $V : S \to \mathbb{R}$, such that $V \models \mathrm{OPT}(\Gamma^{\mu})$
2. If $V \not\models \mathrm{OPT}(\Gamma)$
   then $\mu := \mathrm{Improve}(\mu, V)$; goto 1.

Correctness of 1-player Strategy Improvement

Improvement Lemma

If $V \models \mathrm{Opt}(\Gamma^{\mu})$,

$\mu' = \mathrm{Improve}(\mu, V)$, and

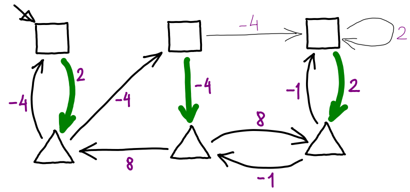
$V' \models \mathrm{Opt}(\Gamma^{\mu'})$

then $\boxed{V' \leq V, \text{ and}}$
$\boxed{V' < V \ \text{ if } \mu' \neq \mu}$

Termination Lemma

Strategy improvement terminates (in $\leq |\Pi_{Min}|$ steps) and returns $V \models \mathrm{Opt}(\Gamma)$.

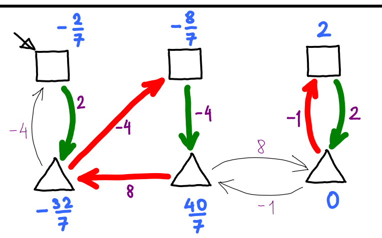
2-player Strategy Improvement



$\delta = \frac{1}{2}$

0. Pick $\chi \in \Pi_{Max}$	
1. Find $V : S \to \mathbb{R}$, such that $V \models \mathrm{Opt}(\Gamma^{\chi})$	Compute the best response μ to χ for Min
2. If $V \not\models \mathrm{Opt}(\Gamma)$ then $\chi := \mathrm{Improve}(\chi, V)$; goto 1.	Improve locally w.r.t. the best response

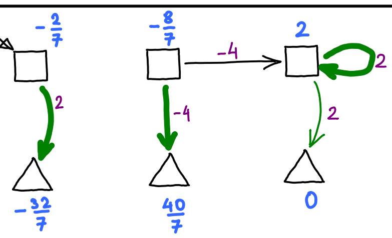
2-player Strategy Improvement



$\delta = \frac{1}{2}$

0. Pick $\chi \in \Pi_{Max}$	
1. Find $V : S \to \mathbb{R}$, such that $V \models \mathrm{Opt}(\Gamma^{\chi})$	**Compute the best response μ to χ for Min**
2. If $V \not\models \mathrm{Opt}(\Gamma)$ then $\chi := \mathrm{Improve}(\chi, V)$; goto 1.	Improve locally w.r.t. the best response

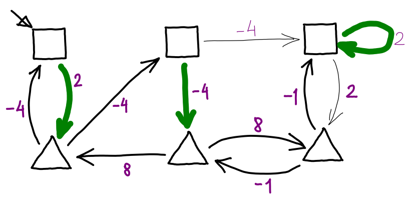
2-player Strategy Improvement



$\delta = \frac{1}{2}$

0. Pick $\chi \in \Pi_{Max}$	
1. Find $V : S \to \mathbb{R}$, such that $V \models \mathrm{Opt}(\Gamma^{\chi})$	Compute the best response μ to χ for Min
2. If $V \not\models \mathrm{Opt}(\Gamma)$ then $\chi := \mathrm{Improve}(\chi, V)$; goto 1.	**Improve locally w.r.t. the best response**

2-player Strategy Improvement



$\delta = \frac{1}{2}$

0. Pick $\chi \in \Pi_{Max}$	
1. Find $V : S \to \mathbb{R}$, such that $V \models \mathrm{Opt}(\Gamma^{\chi})$	**Compute the best response μ to χ for Min**
2. If $V \not\models \mathrm{Opt}(\Gamma)$ then $\chi := \mathrm{Improve}(\chi, V)$; goto 1.	Improve locally w.r.t. the best response

Correctness of 2-player Strategy Improvement

Improvement Lemma

If $V \models \mathrm{Opt}(\Gamma^{\chi})$,

$\chi' = \mathrm{Improve}(\chi, V)$, and

$V' \models \mathrm{Opt}(\Gamma^{\chi'})$

then $V' \geqslant V$, and

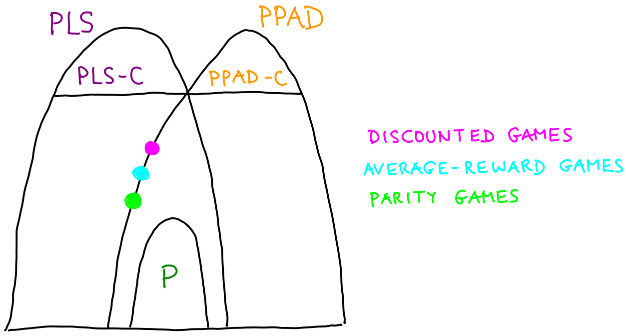
$V' > V$ if $\chi' \neq \chi$.

Termination Lemma

Strategy improvement terminates (in $\leq |\Pi_{Max}|$ steps) and returns $V \models \mathrm{Opt}(\Gamma)$.

Corollary Computing the value of discounted, average-reward, and parity games is in PLS

Computational complexity of solving infinite games



Bellman equations and strategy improvement for average-reward games

Gain-bias Bellman equations for average-reward games:

$$G_s = \begin{cases} \max_{(s,t)\in E} G_t & \text{if } s \in S_{Max} \\ \min_{(s,t)\in E} G_t & \text{if } s \in S_{Max} \end{cases}$$

$$B_s = \begin{cases} \max_{(s,t)\in E} \{ r_{(s,t)} - G_s + B_t : G_t = G_s \} & \text{if } s \in S_{Max} \\ \min_{(s,t)\in E} \{ r_{(s,t)} - G_s + B_t : G_t = G_s \} & \text{if } s \in S_{Min} \end{cases}$$

Bellman equations and strategy improvement for parity games

"Discrete" Bellman equations for parity games.

Thm Strategy improvement for 1-player parity games terminates in **polynomial time**.

Thm [Friedman '09]
Strategy improvement for 2-player parity games requires **exponential time**

Plan

II Quantitative games

1. Motivating examples
2. Average-reward and discounted games
3. Dynamic programming: value iteration
4. Local search: strategy improvement
5. **Mathematical programming: linear complementarity problems**

Linear Complementarity Problem

Given: $M \in \mathbb{R}^{n\times n}$
$q \in \mathbb{R}^n$

Find: $z, w \in \mathbb{R}^n$

such that

linear $\begin{cases} z \geq 0 \\ w \geq 0 \\ w = Mz + q \end{cases}$

complementarity $\{ z \perp w$ i.e., $z^T \cdot w = 0$

Fact
If $q \geq 0$
then $(\underset{z}{0}, \underset{w}{q}) \models \mathrm{LCP}(M, q)$

Complementary cones of LCP

$$\boxed{z \geq 0 \quad \perp \quad w = Mz + q \geq 0}$$

iff

$$\boxed{z \geq 0 \perp w \geq 0 \quad \text{and} \quad q = -Mz + Iw}$$

iff

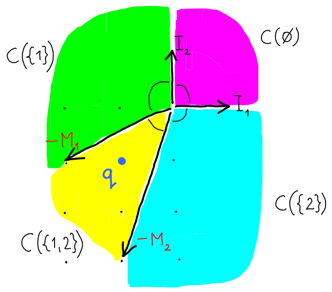
$$\boxed{q \in C(\alpha) = \text{cone}\left(\{-M_k : k \in \alpha\} \cup \{I_k : k \notin \alpha\} \right)}$$

for some $\alpha = \{k : w_k = 0\} \subseteq \{1, \dots, n\}$

Complementary cones of a matrix

$z \geq 0 \perp w \geq 0$ and $q = -Mz + Iw$

$$M = \begin{array}{cc} M_1 & M_2 \\ \end{array} \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \qquad q = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$



$q \in C(\{1,2\})$

$\Downarrow$

$(z_1, z_2, 0, 0) \models LCP(M, q)$

where $q = -Mz$

P-matrices

Def $M \in \mathbb{R}^{n\times n}$ is a P-matrix if all its principal minors are positive

Thm M is a P-matrix iff $LCP(M,q)$ has a unique solution for every $q \in \mathbb{R}^n$

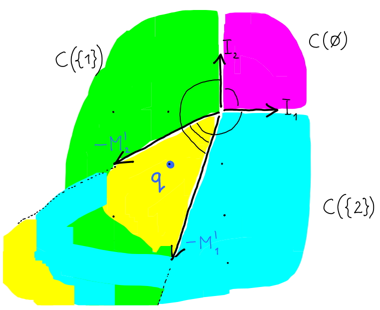
Example $M = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$ $M' = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix}$

- M is a P-matrix: $\det(M_{11}) = 2$, $\det(M_{22}) = 3$, $\det(M) = 5$
- M' is not a P-matrix: $\det(M') = -5$

Complementary cones of a non P-matrix

$z \geq 0 \perp w \geq 0$ and $q = -Mz + Iw$

$$M' = \begin{array}{cc} M'_1 & M'_2 \\ \end{array} \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix} \qquad q = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$



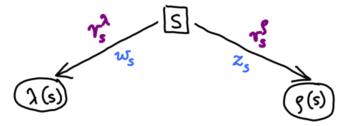
$q \in C(\{1,2\})$

$q \in C(\{1\})$

$q \in C(\{2\})$

Bellman equations and comlementarity

$$v_s = \max\left\{ r_s^\lambda + \delta \cdot v_{\lambda(s)},\ r_s^\varrho + \delta \cdot v_{\varrho(s)} \right\}$$



slacks

$$v_s = w_s + r_s^\lambda + \delta \cdot v_{\lambda(s)}$$
$$v_s = z_s + r_s^\varrho + \delta \cdot v_{\varrho(s)}$$

$z_s, w_s \geq 0$ $z_s \cdot w_s = 0$ complementarity

Bellman equations and comlementarity

$s \in S_{Max}$: $v_s = \max\left\{ r_s^\lambda + \delta \cdot v_{\lambda(s)},\ r_s^\varrho + \delta \cdot v_{\varrho(s)} \right\}$

$s \in S_{Min}$: $v_s = \min\left\{ r_s^\lambda + \delta \cdot v_{\lambda(s)},\ r_s^\varrho + \delta \cdot v_{\varrho(s)} \right\}$

Replace max/min with slacks and complementarity

$s \in S_{Max}$:
$$v_s = w_s + r_s^\lambda + \delta \cdot v_{\lambda(s)}$$
$$v_s = z_s + r_s^\varrho + \delta \cdot v_{\varrho(s)}$$

slacks

$s \in S_{Min}$:
$$v_s = -w_s + r_s^\lambda + \delta \cdot v_{\lambda(s)}$$
$$v_s = -z_s + r_s^\varrho + \delta \cdot v_{\varrho(s)}$$

$s \in S$: $w_s \geq 0 \perp z_s \geq 0$ complementarity

Reduction to LCP: rewrite

$s \in S_{Max}$:
$$v_s = w_s + r_s^\lambda + \delta \cdot v_{\lambda(s)}$$
$$v_s = z_s + r_s^\varrho + \delta \cdot v_{\varrho(s)}$$

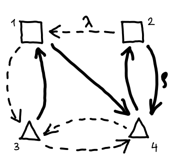
slacks

$s \in S_{Min}$:
$$v_s = -w_s + r_s^\lambda + \delta \cdot v_{\lambda(s)}$$
$$v_s = -z_s + r_s^\varrho + \delta \cdot v_{\varrho(s)}$$

$\rightsquigarrow$

$$(\hat{I} - \delta \cdot \hat{T}^\lambda)\, v = w + \hat{I} r^\lambda$$
$$(\hat{I} - \delta \cdot \hat{T}^\varrho)\, v = z + \hat{I} r^\varrho$$

Example



$$\hat{I} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

$$\hat{T}^\varrho = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

Reduction to LCP: eliminate v

$$\boxed{\begin{aligned}(\hat{I}-\delta\cdot\widehat{T^{\lambda}})\,v &= w+\hat{I}\,r^{\lambda}\\ (\hat{I}-\delta\cdot\widehat{T^{\varrho}})\,v &= z+\hat{I}\,r^{\varrho}\end{aligned}}$$

Eliminate v: $\quad w+\hat{I}\,r^{\lambda} = \underbrace{(\hat{I}-\delta\cdot\widehat{T^{\lambda}})\cdot(\hat{I}-\delta\cdot\widehat{T^{\varrho}})^{-1}}_{M}\cdot(z+\hat{I}\,r^{\varrho})$

$$\boxed{\begin{aligned}&w = Mz+q\\ &w\geq 0 \perp z\geq 0\end{aligned}}$$

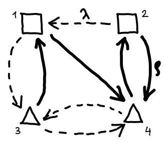
$$\boxed{\begin{aligned}M &= (\hat{I}-\delta\cdot\widehat{T^{\lambda}})\cdot(\hat{I}-\delta\cdot\widehat{T^{\varrho}})^{-1}\\ q &= M(\hat{I}\,r^{\varrho})-(\hat{I}\,r^{\lambda})\end{aligned}}$$

Reduction to LCP: eliminate v

$$\boxed{\begin{aligned}&w = Mz+q\\ &w\geq 0 \perp z\geq 0\end{aligned}}$$

$$\boxed{\begin{aligned}M &= (\hat{I}-\delta\cdot\widehat{T^{\lambda}})\cdot(\hat{I}-\delta\cdot\widehat{T^{\varrho}})^{-1}\\ q &= M(\hat{I}\,r^{\varrho})-(\hat{I}\,r^{\lambda})\end{aligned}}$$

Example



$$\hat{I}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&-1\end{pmatrix}\qquad \widehat{T^{\varrho}}=\begin{pmatrix}0&0&0&1\\0&0&0&1\\-1&0&0&0\\0&-1&0&0\end{pmatrix}$$

$$(\hat{I}-\delta\cdot\widehat{T^{\varrho}})=\begin{pmatrix}1&0&0&-\delta\\0&1&0&-\delta\\\delta&0&-1&0\\0&\delta&0&-1\end{pmatrix}$$

Strictly diagonally dominant matrices

Fact [Levy-Desplanques]
If $A\in\mathbb{R}^{n\times n}$ is strictly diagonally dominant, then A is non-singular.

Fact $(\hat{I}-\delta\cdot\widehat{T^{\lambda}})$ and $(\hat{I}-\delta\cdot\widehat{T^{\varrho}})$ are s.d.d.

Corollary $M=(\hat{I}-\delta\cdot\widehat{T^{\lambda}})\cdot(\hat{I}-\delta\cdot\widehat{T^{\varrho}})^{-1}$ is well-defined

P-matrices of the form $B\cdot C^{-1}$

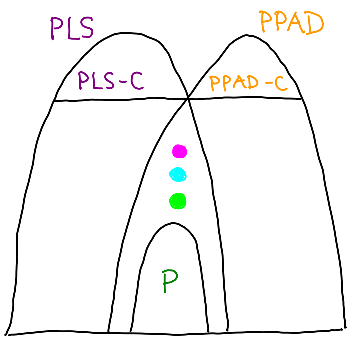
Thm [Johnson-Tsatsomeros '95]
Let $M=B\cdot C^{-1}$, where $B,C\in\mathbb{R}^{n\times n}$

Then, M is a P-matrix if $T\cdot B+(I-T)\cdot C$ is non-singular for all $T\in[0,I]$

Corollary

- $\boxed{M=(\hat{I}-\delta\cdot\widehat{T^{\lambda}})\cdot(\hat{I}-\delta\cdot\widehat{T^{\varrho}})^{-1} \text{ is a P-matrix.}}$
- Computing the value of discounted, average-reward, and parity games is in PPAD

Computational complexity of solving infinite games



DISCOUNTED GAMES
AVERAGE-REWARD GAMES
PARITY GAMES

What is q?

Lemma
Let v^{ϱ} be the vector of values of strategy pair ϱ, i.e.,

$$\boxed{v^{\varrho}=r^{\varrho}+\delta\cdot T^{\varrho}\cdot v^{\varrho}}$$

Then:

$$\boxed{q=\hat{I}\Big(\underbrace{(r^{\varrho}+\delta\cdot T^{\varrho}\cdot v^{\varrho})}_{\text{take "right" edge and then follow }\varrho}-\underbrace{(r^{\lambda}+\delta\cdot T^{\lambda}\cdot v^{\varrho})}_{\text{take "left" edge and then follow }\varrho}\Big)}$$

Algorithm 1: Strategy improvement revisited

Def. $s \in S_{Max}$ is switchable (for strategy pair ϱ) if

$$\boxed{r_s^{\varrho} + \delta \cdot v^{\varrho}_{\varrho(s)} < r_s^{\lambda} + \delta \cdot v^{\varrho}_{\lambda(s)}}$$

1. Start with arbitrary ϱ, λ.
2. <u>While</u> **there is** a switchable vertex (for ϱ) <u>do</u>
3. Find ϱ' such that:
   a) $\varrho' \upharpoonright S_{Max} = \varrho \upharpoonright S_{Max}$
   b) **no** $s \in S_{Min}$ is switchable for ϱ'

   $\varrho' \upharpoonright S_{Min}$ is a best response to $\varrho \upharpoonright S_{Max}$
4. Switch all (or some) switchable vertices in S_{Max}

Algorithm 2: Murty's "least-index" method

1. Fix a permutation ("indexing") of states.
2. <u>While</u> there is a switchable vertex <u>do</u>
3. Switch the switchable vertex with **least index**.

Thm Murty's algorithm terminates (in $\leqslant 2^n$ steps).

"Nested" strategy improvement: $^1\triangle$ $^2\triangle$ $^3\triangle$ $^4\square$ $^5\square$

New algorithm: $^1\square$ $^2\triangle$ $^3\triangle$ $^4\square$ $^5\triangle$

Algorithm 3: Cottle-Danzig

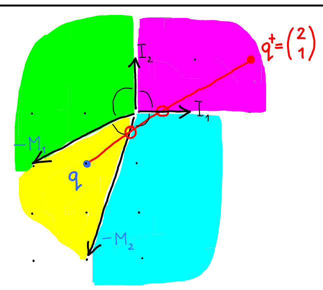
1. <u>For</u> $s = 1, 2, \dots, n$ <u>do</u>
2. <u>If</u> s is switchable <u>then</u>
3. Drive r_s^{ϱ} until s becomes **indifferent**
4. <u>While</u> driving r_s^{ϱ},
   switch a vertex in $\{1, 2, \dots, s-1\}$
   when it becomes **indifferent**
5. Switch s and restore r_s^{ϱ}

Fact Cottle-Danzig terminates (in $\leqslant 2^n$ steps)

Algorithm 4: Lemke

1. Drive r^{λ} ("linearly"), until no vertex is switchable.
2. Drive r^{λ} ("linearly") back, until original r^{λ} is restored.
   <u>While</u> driving r^{λ} back,
   switch vertices when they become indifferent.

$$q = M(\hat{I} r^{\varrho}) - (\hat{I} r^{\lambda})$$



Lemke needs exponential time

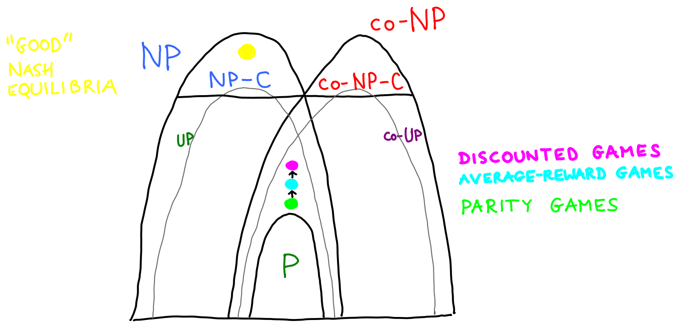
Thm [Savani, von Stengel 2004]
Lemke-Howson algorithm needs exponential time to find a Nash equilibrium

Thm [Fearnley, J., Savani 2009]
Lemke's algorithm needs exponential time to find optimal strategies

Lemke can be fast

Thm [Adler, Meggido 1985]
There is an implementation of Lemke's algorithm that terminates in quadratic number of steps on random linear programs

Computational complexity of solving games



Computational complexity of solving games