Multigrid methods for two player zero-sum stochastic games

Sylvie Detournay

INRIA Saclay and CMAP, École Polytechnique

Soutenance de thèse
Le 25 septembre, 2012

Outline

- Zero-sum two player stochastic game with discounted payoff
  - Dynamic Programing equations
  - Policy iteration and multigrids : AMGπ
  - Numerical results
- Zero-sum two player stochastic game with mean payoff
  - Unichain case
    - Dynamic Programing equations
    - Policy iteration and multigrids : AMGπ
    - Numerical results
  - Multichain case
    - Dynamic Programing equations
    - Policy iteration for multichain
    - Numerical results
- Conclusions

Dynamic programming equation of zero-sum two-player stochastic games

$$v(x) = \max_{a \in \mathcal{A}(x)} \min_{b \in \mathcal{B}(x,a)} \sum_{y \in \mathcal{X}} \gamma P(y|x,a,b) v(y) + r(x,a,b)$$

$$\forall x \in \mathcal{X} \quad \text{(DP)}$$

$\mathcal{X}$ state space
$v(x)$ the value of the game starting at $x \in \mathcal{X}$,
a, b action of the 1st, 2nd player MAX, MIN
$r(x,a,b)$ reward paid by MIN to MAX
$P(y|x,a,b)$ transition probability from x to y given the actions a, b
$\gamma < 1$ discount factor

Value of the game starting in x

$$v(x) = \max_{(a_k)_{k\geq 0}} \min_{(b_k)_{k\geq 0}} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r(X_k, a_k, b_k)\right]$$

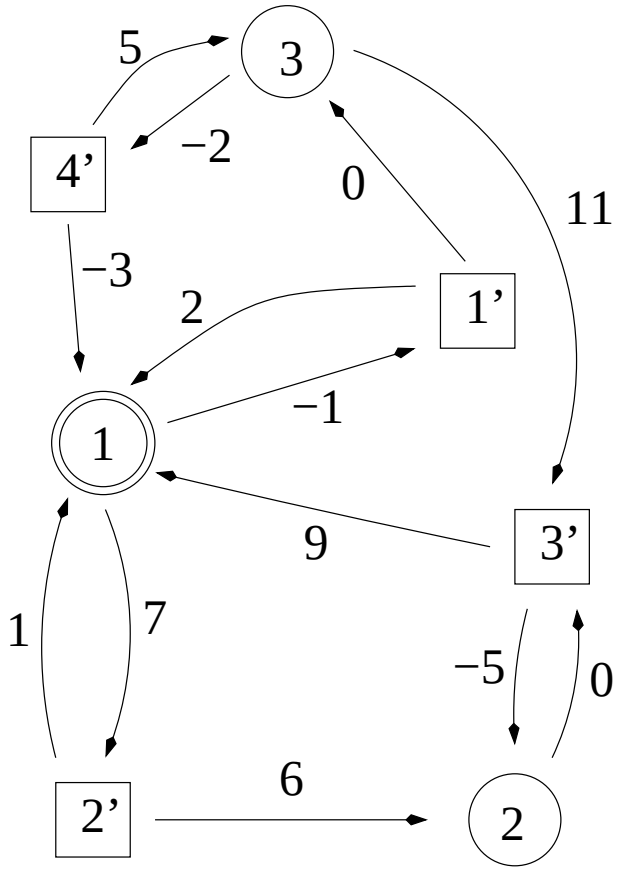
where

$$\begin{cases} a_k = a_k(X_k, b_{k-1}, a_{k-1}, X_{k-1} \cdots) \\ b_k = b_k(X_k, a_k, \cdots) \end{cases}$$

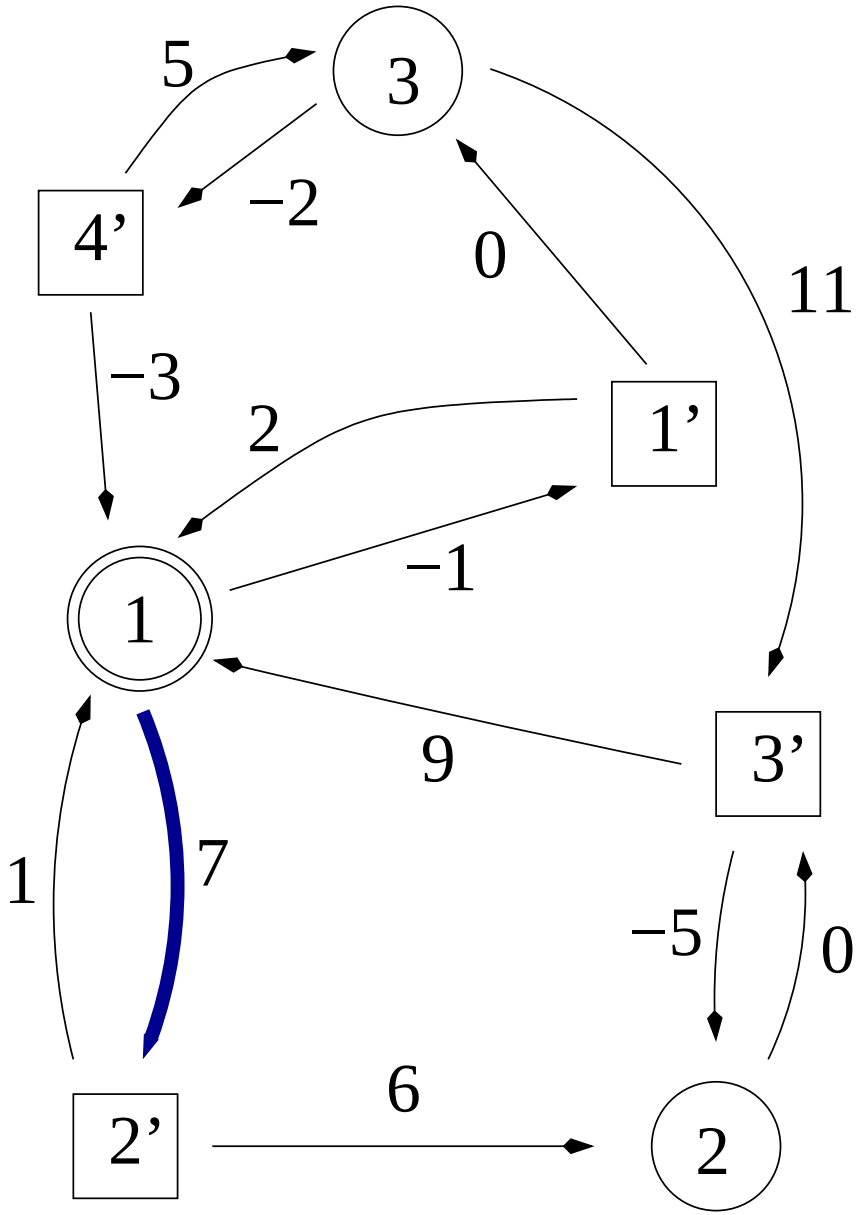
are strategies and the state dynamics satisfies the process X_k

$$P(X_{k+1} = y | X_k = x, a_k = a, b_k = b) = P(y|x, a, b)$$

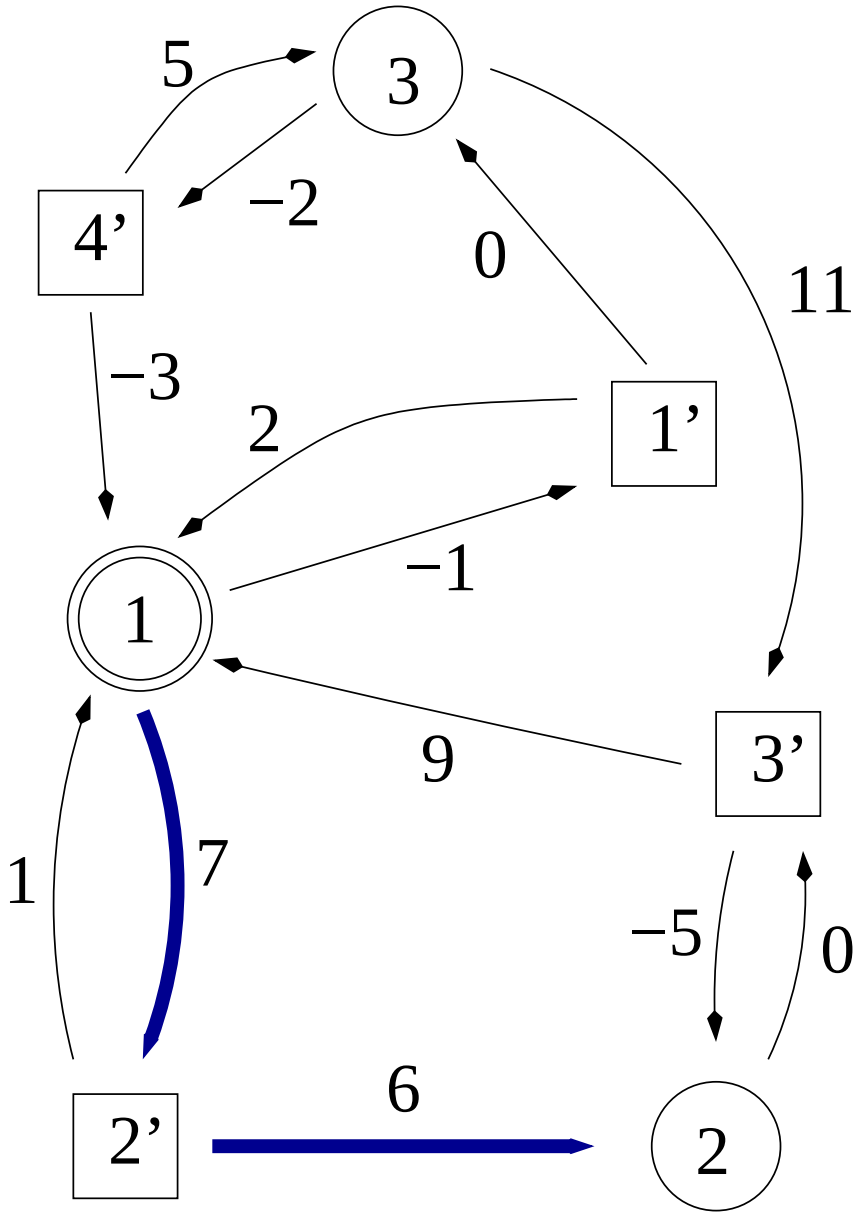
Deterministic zero-sum two-player game



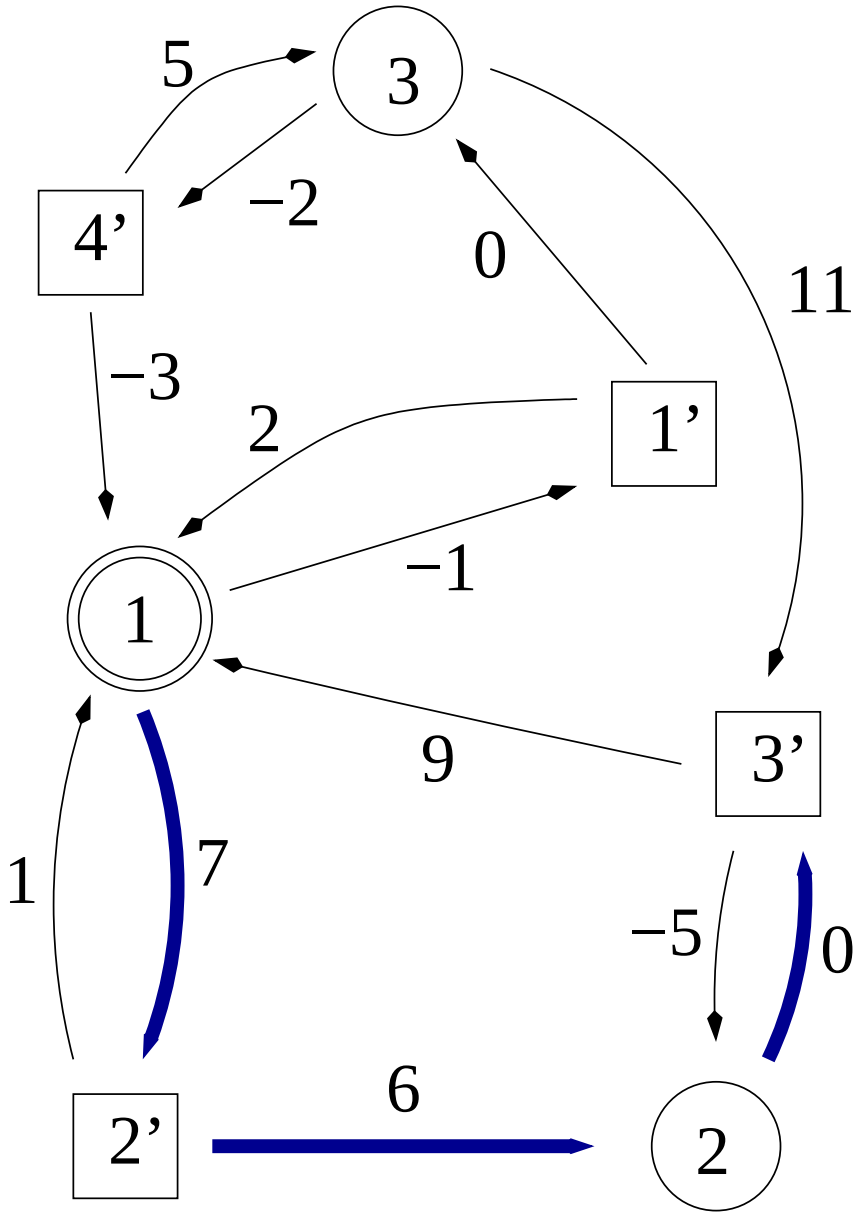
Circles : Max plays
Squares : MIN plays
Weight on the edges : payment made by MIN to MAX



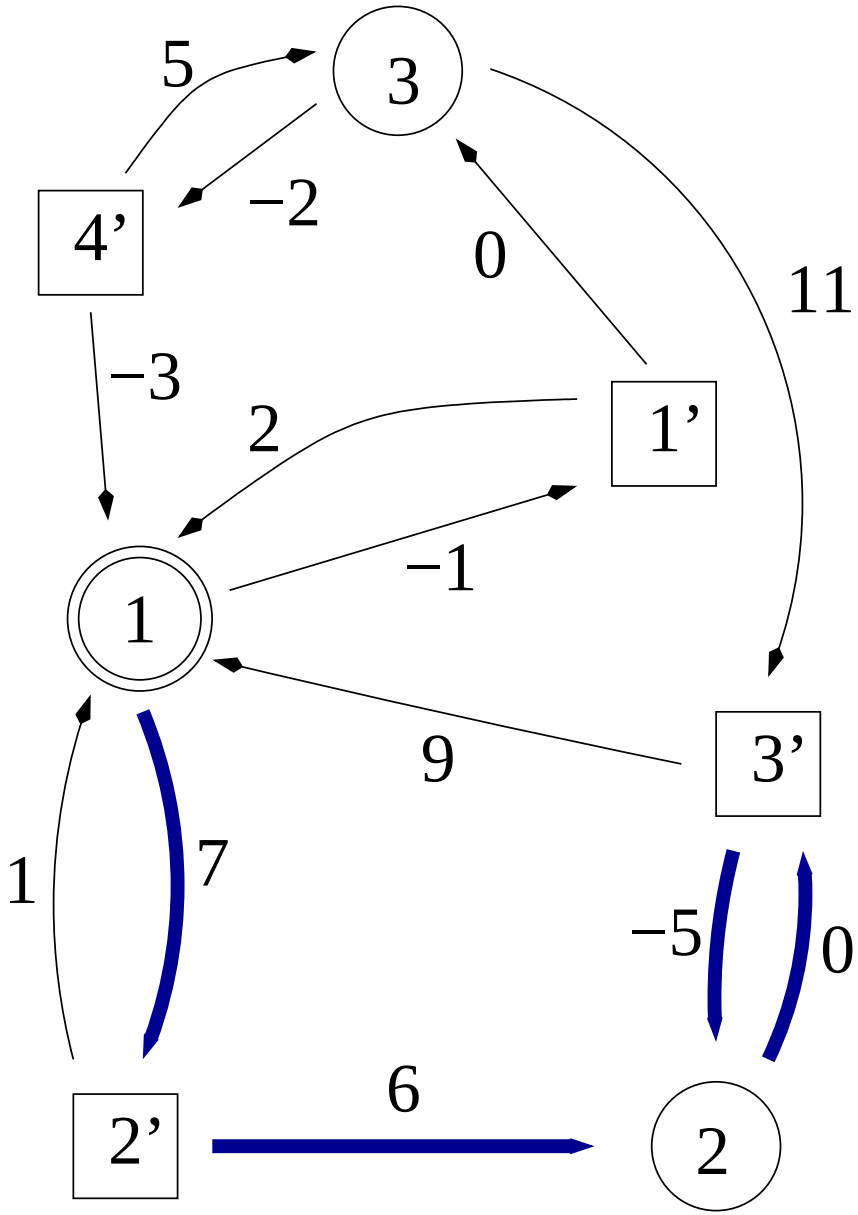
If Max initially moves to $2'$



If Max initially moves to $2'$

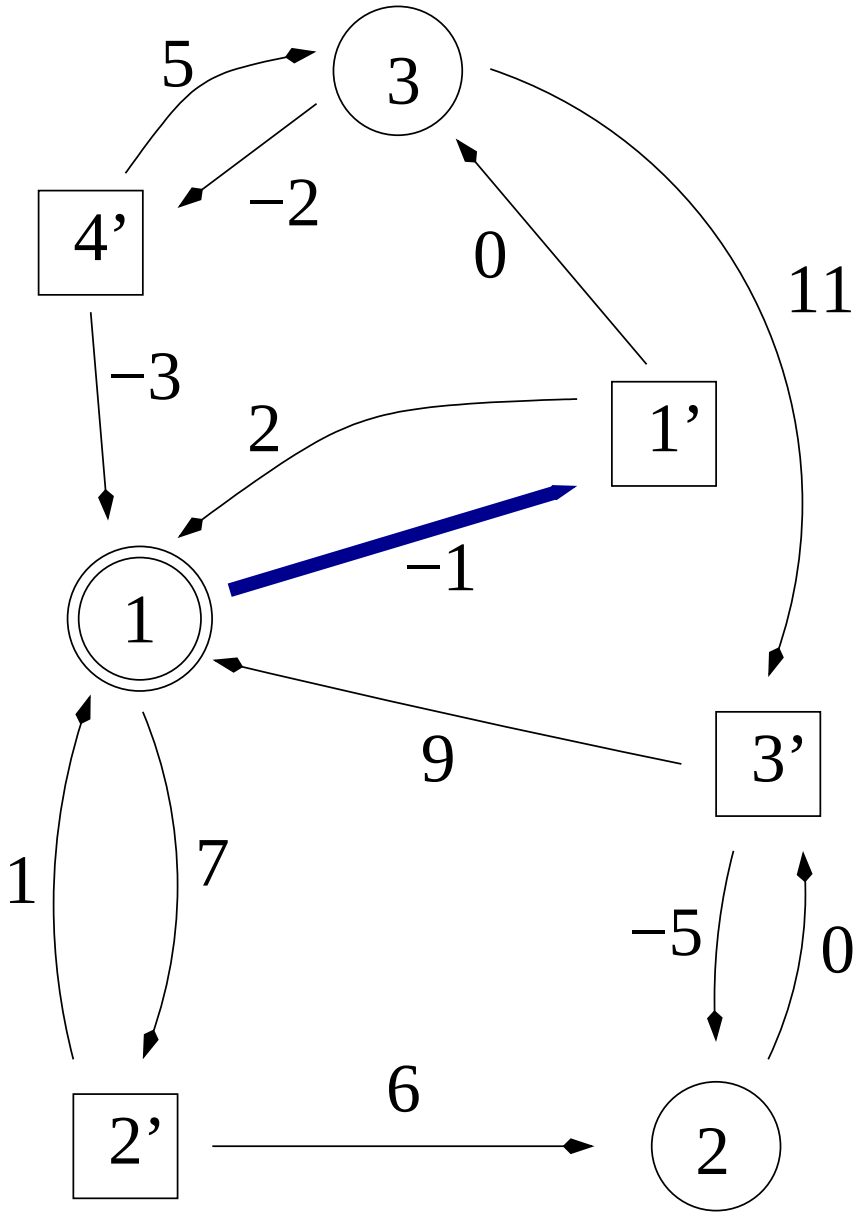


If Max initially moves to $2'$

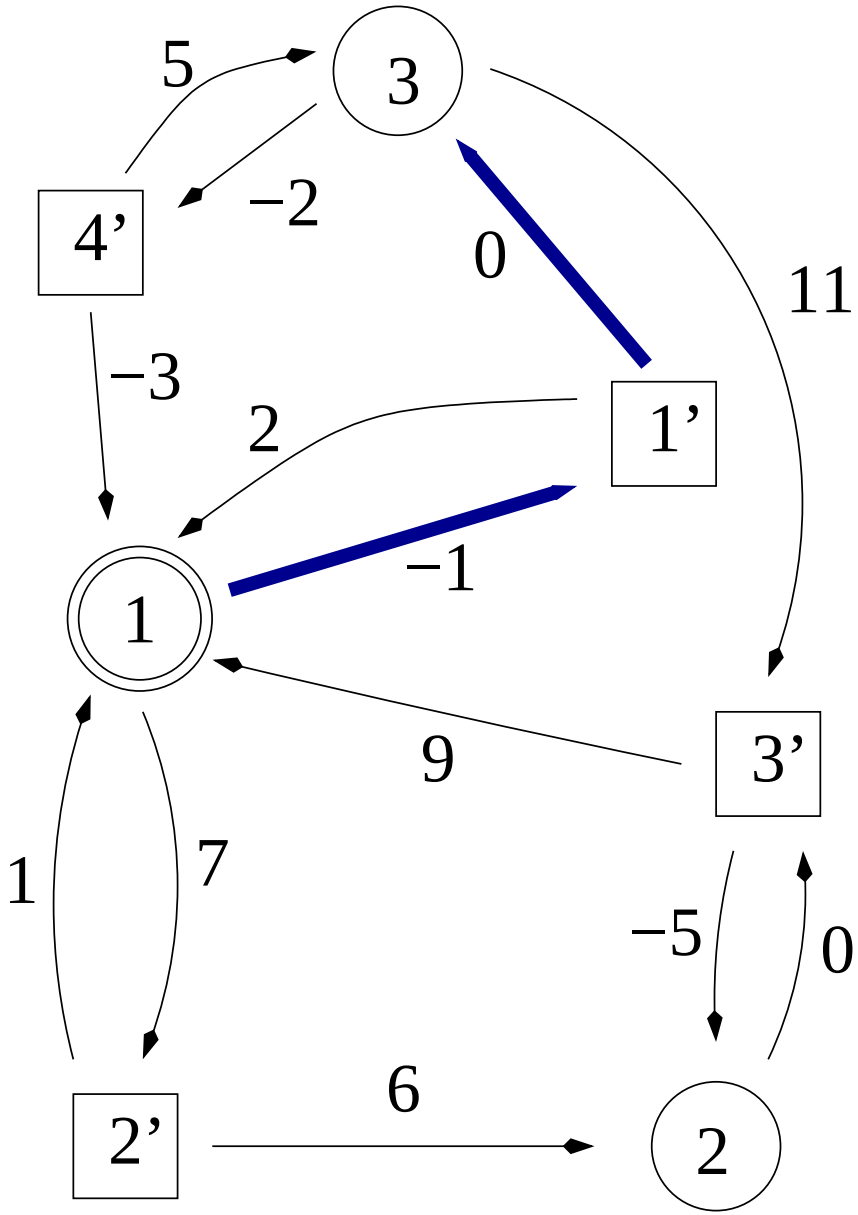


If Max initially moves to $2'$

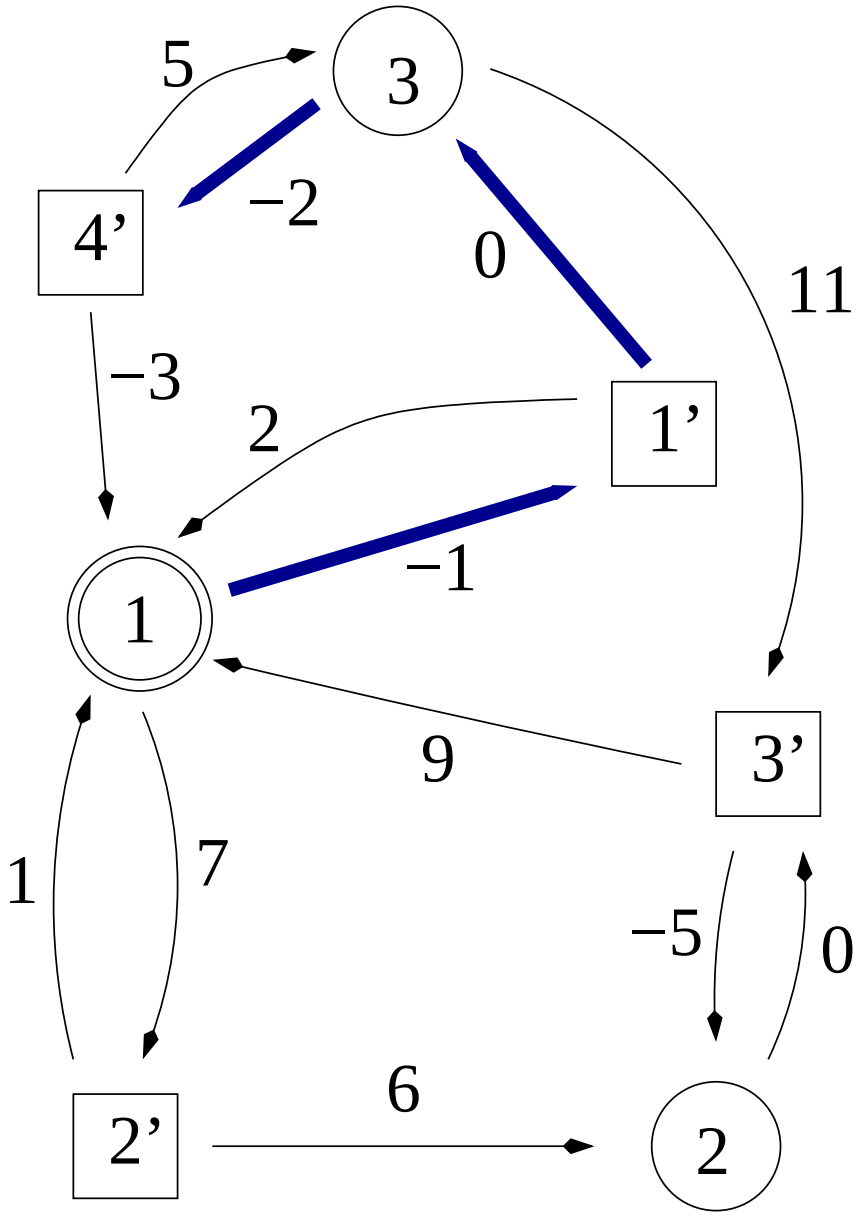
he eventually looses 5 per turn.



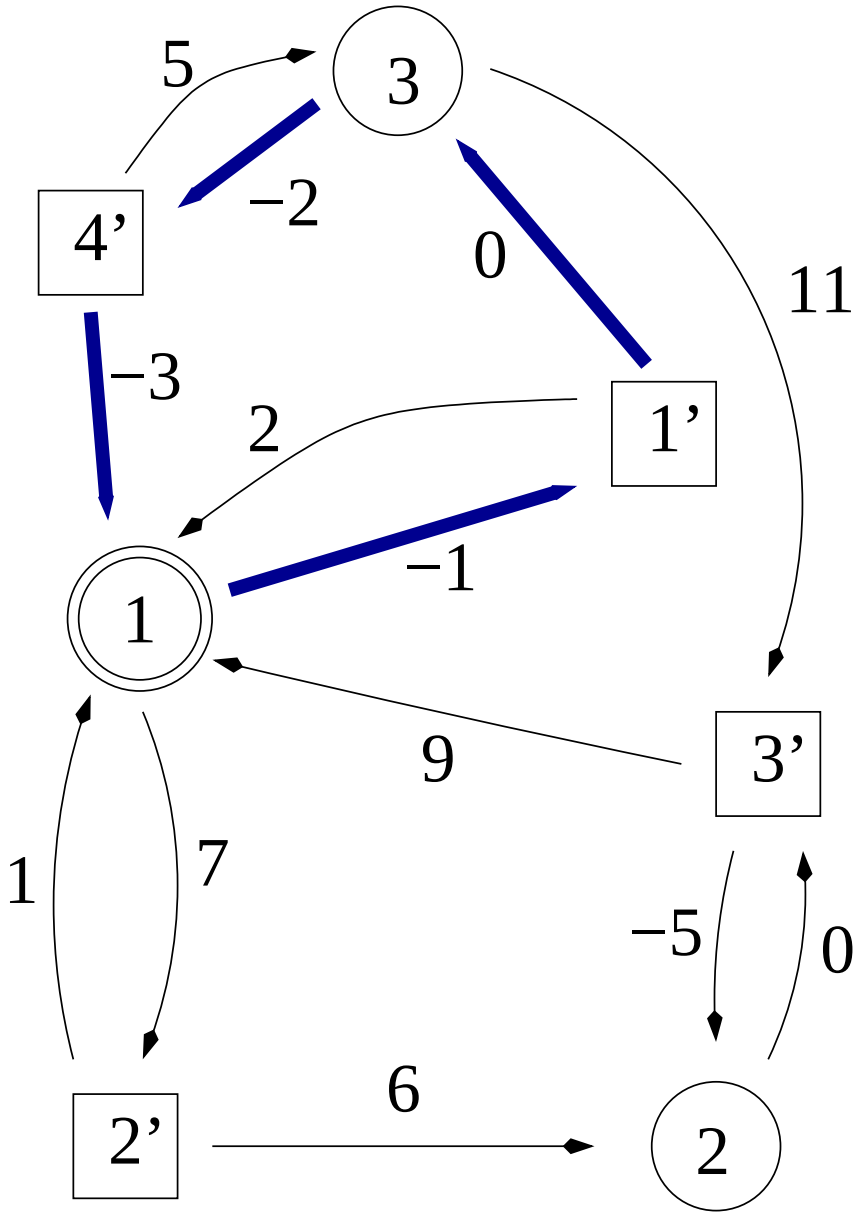
But if Max initially moves to $1'$



But if Max initially moves to $1'$



But if Max initially moves to $1'$



But if Max initially moves to $1'$

he only looses eventually $(1+0+2+3)/2 = 3$ per turn.

Feedback strategies or policy

$$v(x) = \max_{(a_k)_{k\geq 0}} \min_{(b_k)_{k\geq 0}} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r(X_k, a_k, b_k)\right]$$

For $\alpha : x \to \alpha(x) \in \mathcal{A}(x)$ and $\beta : (x, a) \to \beta(x, a) \in \mathcal{B}(x, a)$, the strategies

$$\begin{cases} a_k = \alpha(X_k) \\ b_k = \beta(X_k, a_k) \end{cases}$$

are such that X_k is a Markov Chain with transition matrix $P^{\alpha,\beta}$ where

$$P^{\alpha,\beta}_{xy} := P(y|x, \alpha(x), \beta(x, \alpha(x)))$$

x, y in $\mathcal{X}$.

Dynamic programming operator and optimal policy

$$v(x) = \max_{a \in \mathcal{A}(x)} \min_{b \in \mathcal{B}(x,a)} \underbrace{\sum_{y \in \mathcal{X}} \gamma P(y|x,a,b) v(y) + r(x,a,b)}_{F(v;(x,a),b)} := F(v;x)$$

α policy maximizing (DP)eq for MAX
β policy minimizing $F(v;(x,a),b)$ for MIN

The dynamic programming operator F is monotone and additively sub-homogeneous ($F(\lambda + v) \leq \lambda + F(v)$, $\lambda \geq 0$).

Method to solve (DP) eqs : Policy iteration algorithm [Howard, 60 (1player game)], [Denardo, 67 (2player game)]

Dynamic programming equation of zero-sum two-player stochastic differential games

PDE of Isaacs (or Hamilton-Jacobi-Bellman for one player)

$$-\lambda v(x) + H(x, \frac{\partial v}{\partial x_i}, \frac{\partial^2 v}{\partial x_i \partial x_j}) = 0, \quad x \in \mathcal{X} \tag{I}$$

where

$$H(x, p, K) = \max_{a \in \mathcal{A}(x)} \min_{b \in \mathcal{B}(x,a)} \Big[p \cdot f(x, a, b) + \frac{1}{2} \mathrm{tr}(\sigma(x, a, b)\sigma^T(x, a, b)K) + r(x, a, b) \Big]$$

Discretization with monotone schemes of (I) yields (DP)

Motivation

Solve dynamic programming equations arising from the discretization of Isaacs equations or other DP eq of diffucions (eg varitional inequalities)
applications: pursuit-evasion games, finance,...

Solve large scale zero-sum stochastic games (with discrete state space)
for example, problems arising from the web, problems in verification of programs in computer science, ...

$\rightarrow$ Use policy iteration algorithm where the linear systems involved are solved using AMG

Policy Iteration (PI) Algorithm for games

$$v(x) = \max_{a \in \mathcal{A}(x)} \underbrace{\min_{b \in \mathcal{B}(x,a)} \sum_{y \in \mathcal{X}} \gamma P(y|x,a,b) v(y) + r(x,a,b)}_{F(v;x,a)}$$

Start with $\alpha_0 : x \to \alpha_0(x) \in \mathcal{A}(x)$, apply successively

1. The value v^{k+1} of policy α_k is solution of

$$v^{k+1}(x) = F(v^{k+1}; x, \alpha_k(x)) \quad \forall x \in \mathcal{X}.$$

2. Improve the policy: select α_{k+1} optimal for v^{k+1}:

$$\alpha_{k+1}(x) \in \operatorname*{argmax}_{a \in \mathcal{A}(x)} F(v^{k+1}; x, a) \quad \forall x \in \mathcal{X}.$$

Until $\alpha_{k+1}(x) = \alpha_k(x)\ \forall x \in \mathcal{X}$.

Step 1 is solved by PI

Policy Iteration (PI) for 1-player games (Howard, 60)

Start with $\beta_{k,0}$, apply successively

1. The value $v^{k,s+1}$ of policy $\beta_{k,s}$ is solution of

$$v^{k,s+1} = \gamma P^{\alpha_k,\beta_{k,s}} v^{k,s+1} + r^{\alpha_k,\beta_{k,s}}$$

where $P^{\alpha,\beta}_{xy} := P(y|x,\alpha(x),\beta(x,\alpha(x)))$

2. Improve the policy: find $\beta_{k,s+1}$ optimal for $v^{k,s+1}$

Until $\beta_{k,s+1} = \beta_{k,s}$.

$$PI_{ext}\begin{cases} \alpha_0 & PI_{int}\begin{cases} \beta_{0,0} \\ \vdots \\ \beta_{0,s} \end{cases} \\ \vdots & \\ \alpha_k & \end{cases}$$

$(v^k)_{k\geq 1} \nearrow$ non decreasing (MAX player)
$(v^{k,s})_{s\geq 1} \searrow$ non increasing (MIN player)

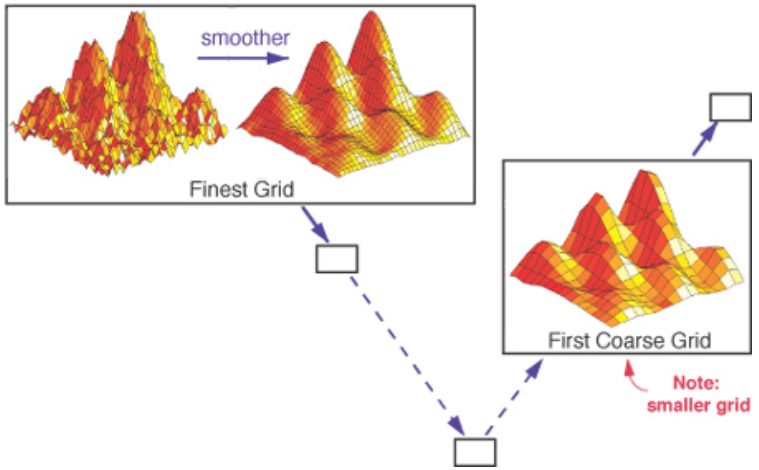
PI stops after a finite time when sets of actions are finite

Internal loop (1player game): PI $\approx$ Newton algorithm where differentials are replaced by superdifferentials of the (DP) operator

External loop (2player game): PI $\approx$ Newton algorithm where the (DP) operator is approached by below by piecewise affine and concave maps

$\rightarrow$ expect super linear convergence in good cases

MultiGrids for a linear system $Av = b$



- PDE is discretized on a regular grid with n nodes (= finest grid)
  Define a coarse grid with less nodes by tacking even nodes

- Solving phase : (two grids)
  $v \leftarrow$ apply ν_1 relaxations on the fine level to v
  $v \leftarrow v + Iw$ where w is solution of

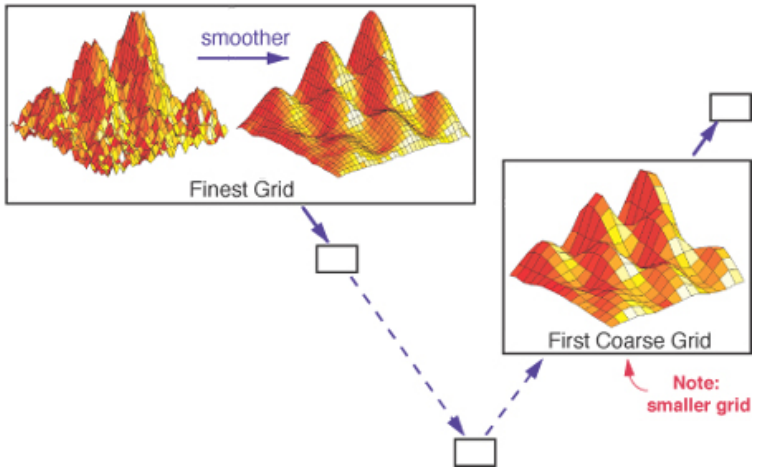
$$RAIw = R(b - Av) \quad \text{(on the coarse grid)}$$

  $v \leftarrow$ apply ν_2 relaxations on the fine level to v

eg relaxation - Jacobi: $v \leftarrow D^{-1}(b - (L + U)v)$ with $A = D + L + U$
when applied recursively $\rightarrow$ V-cycle, W-cycle ...

AMG for a linear system $Av = b$

- Setup phase:
  construct "grids" based on the elements of matrix A
  define interpolation $(I)_{ij} \approx \frac{A_{ij}}{somefactor}$,
  restriction $R = I^T$



- Solving phase : (two grids)
  $v \leftarrow$ apply ν_1 relaxations on the fine level to v
  $v \leftarrow v + Iw$ where w is solution of

$$RAIw = R(b - Av) \quad \text{(on the coarse grid)}$$

  $v \leftarrow$ apply ν_2 relaxations on the fine level to v

eg relaxation - Jacobi: $v \leftarrow D^{-1}(b - (L + U)v)$ with $A = D + L + U$
when applied recursively $\rightarrow$ V-cycle, W-cycle ....

AMGπ

Combine PI for two-player games and AMG:
Apply AMG to $v = \gamma P v + r$ in the internal loop of PI

$\rightarrow$ AMGπ

$$PI_{ext} \begin{cases} \alpha_0 & PI_{int} \begin{cases} \beta_{0,0} & AMG \begin{cases} v^{0,0,0} \\ \vdots \\ v^{0,0,m} \end{cases} \\ \vdots \\ \beta_{0,s} \end{cases} \\ \vdots \\ \alpha_k \end{cases}$$

Previous works in stochastic control (one player games):
MG + PI in [Hoppe,86-87][Akian, 88-90]
AMG + learning methods [Ziv and Shinkin, 05]
$\rightarrow$ two player games never considered

Example on a Isaacs equations

Dynamic programming equation

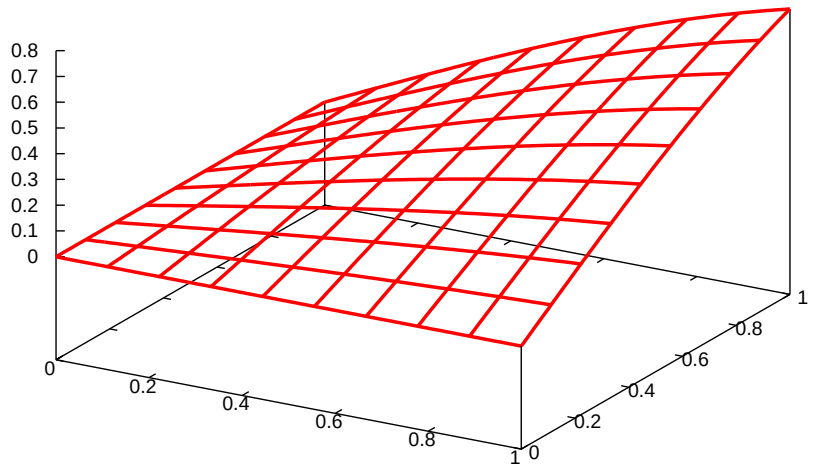
$$\begin{cases} \Delta v(x) + \|\nabla v(x)\|_2 - 0.5\,\|\nabla v(x)\|_2^2 + f(x) = 0 & x \in \mathcal{X} \\ v(x) = g(x) & x \in \partial\mathcal{X} \end{cases}$$

where

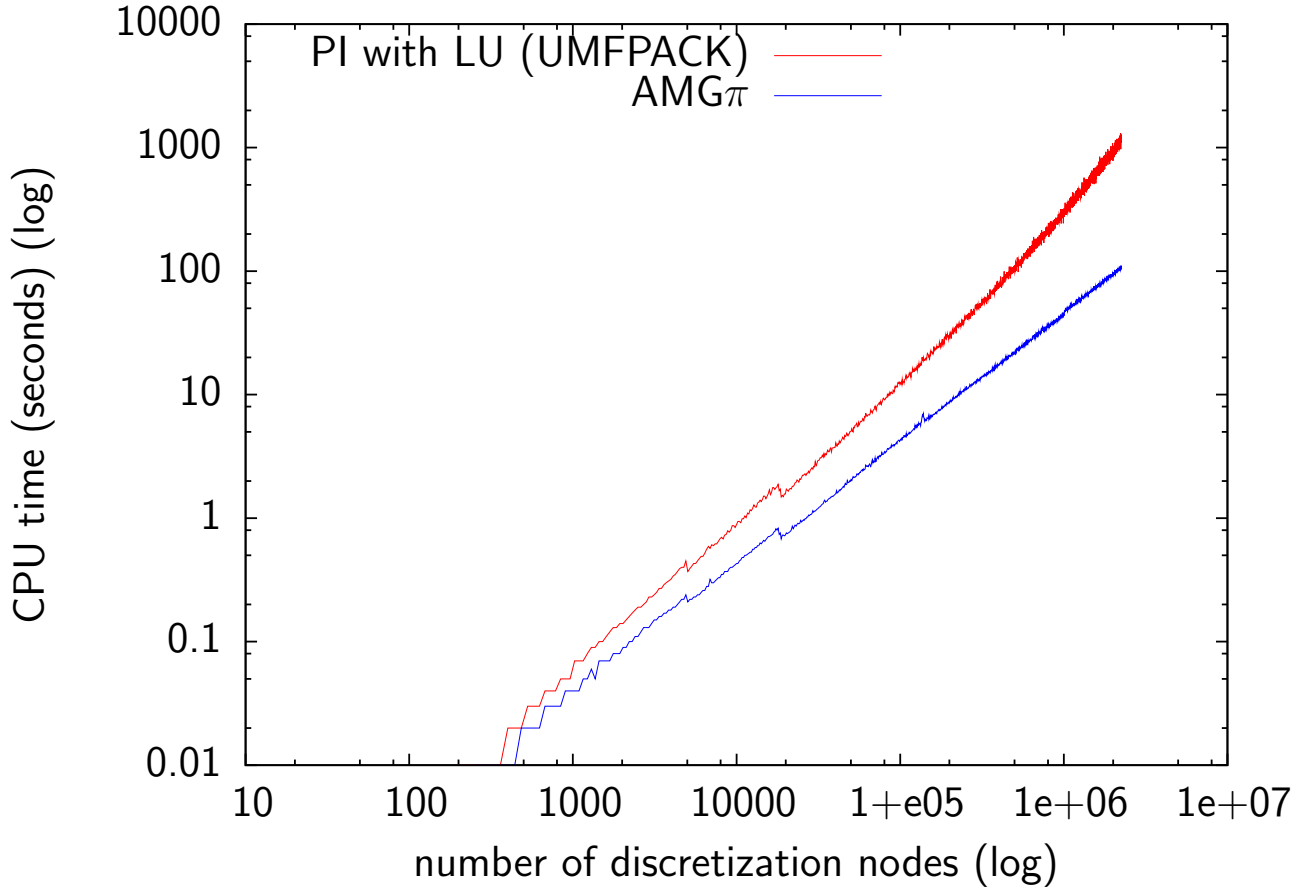
$$\|\nabla v(x)\|_2 = \max_{\|a\|_2 \leq 1} (a \cdot \nabla v(x))$$

$$-\frac{\|\nabla v(x)\|_2^2}{2} = \min_b (b \cdot \nabla v(x) + \frac{\|b\|_2^2}{2})$$

with $v(x_1, x_2) = sin(x_1) \times sin(x_2)$ on $\mathcal{X} = [0,1] \times [0,1]$



AMGπ versus PI with LU



For the 100 problems of finest discretization:

slope ≈ 1.04 for AMGπ, slope ≈ 1.85 for PI with LU.

About 6 linear system solved for each problem, size from 5^2 to 1500^2.

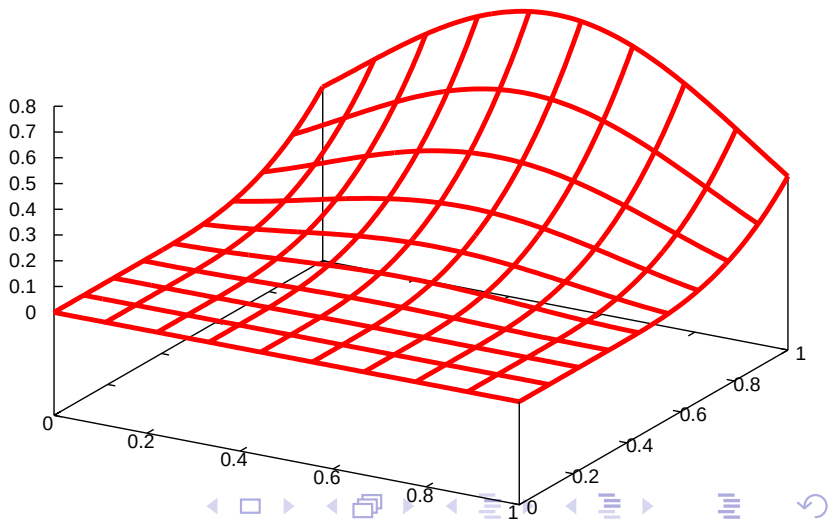
Variational inequalities problem (VI)

Optimal stopping time for first player

$$\begin{cases} max\left[0.5\Delta v(x) - 0.5\left\|\nabla v(x)\right\|_2^2 + f(x), \phi(x) - v(x)\right] = 0 & x \in \mathcal{X} \\ v(x) = u(x) & x \in \partial\mathcal{X} \end{cases}$$

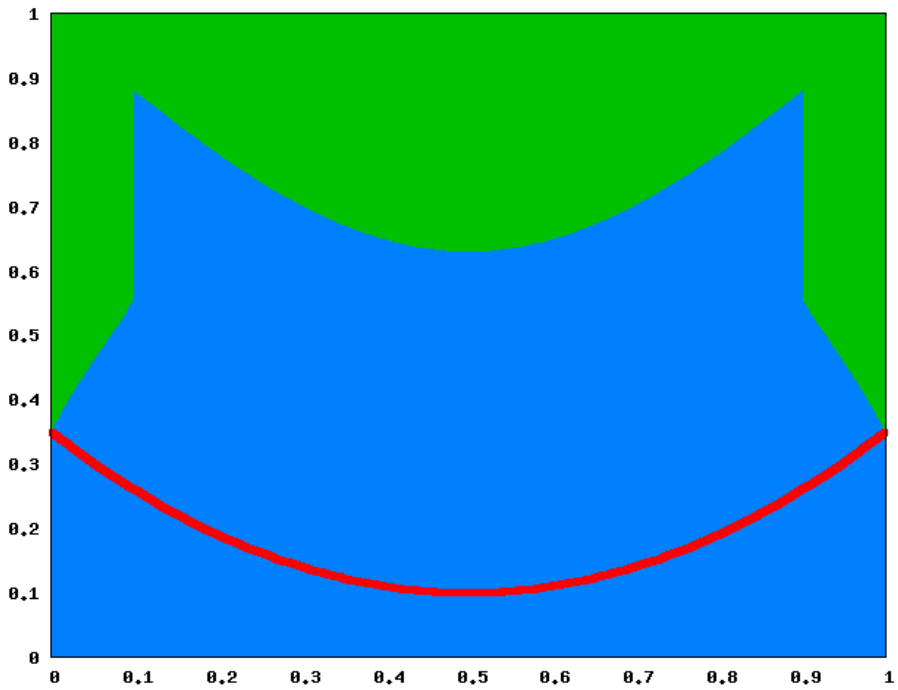
MAX chooses between play or stop ($\sharp\mathcal{A}(x) = 2$) and receives ϕ when he stops
MIN leads $\left\|\nabla v\right\|_2^2$

with $\phi = 0$ and solution v on $\mathcal{X} = [0,1] \times [0,1]$ given by



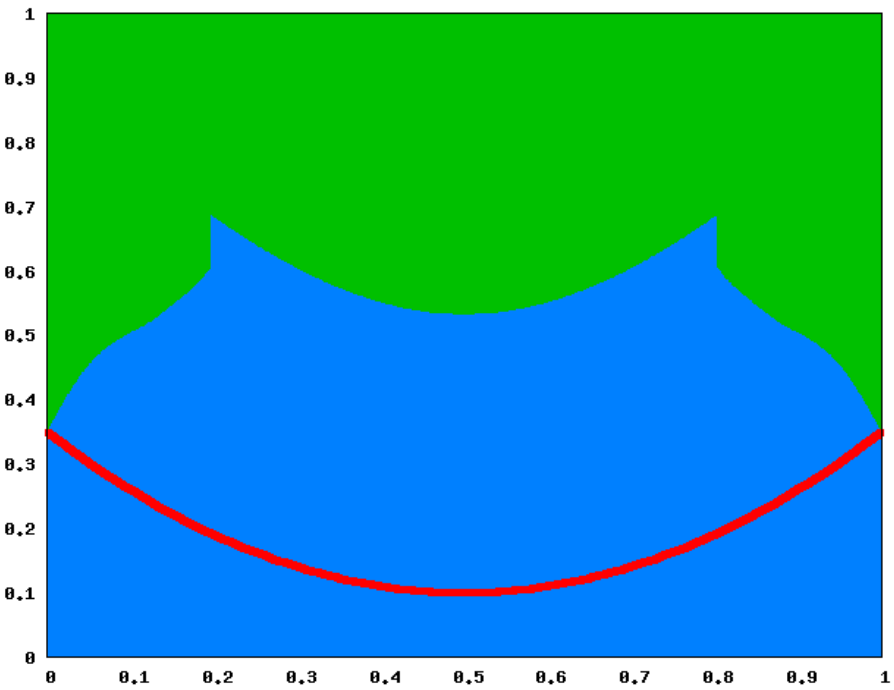
VI with 129×129 points grid

iterations $= 100$



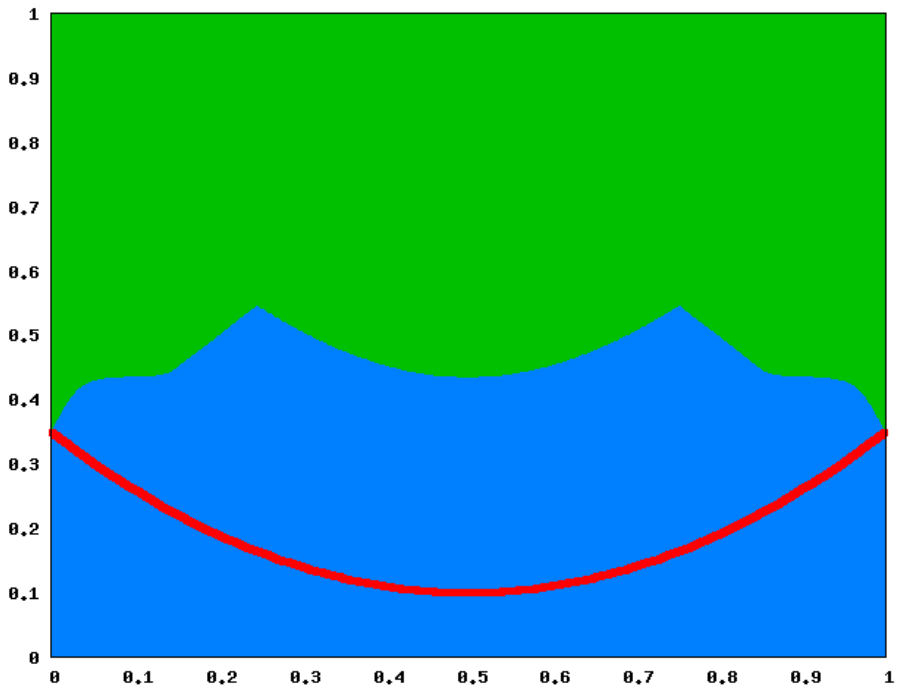
VI with 129×129 points grid

iterations $= 200$



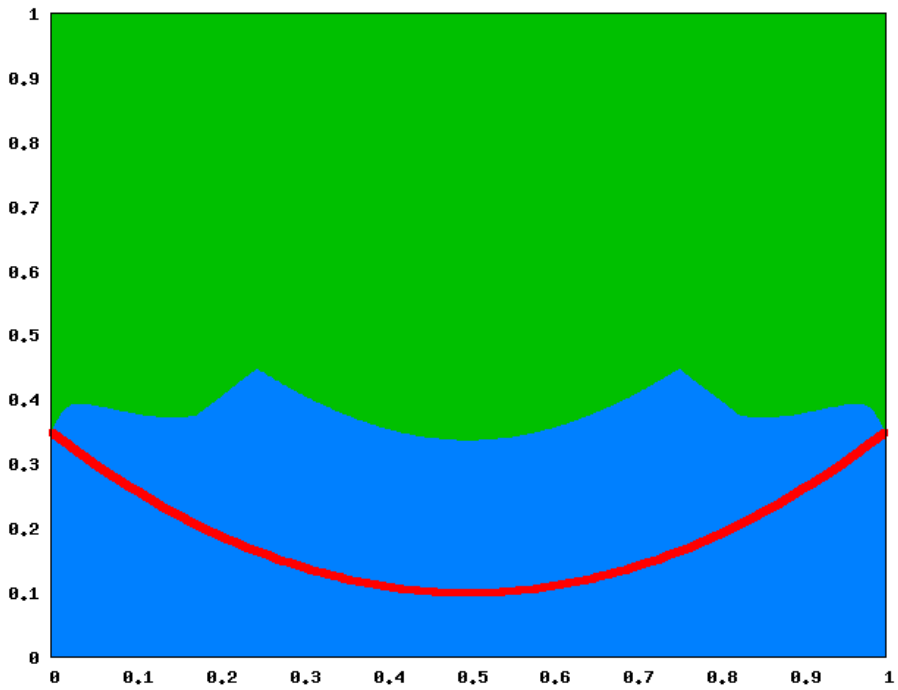
VI with 129×129 points grid

iterations $= 300$



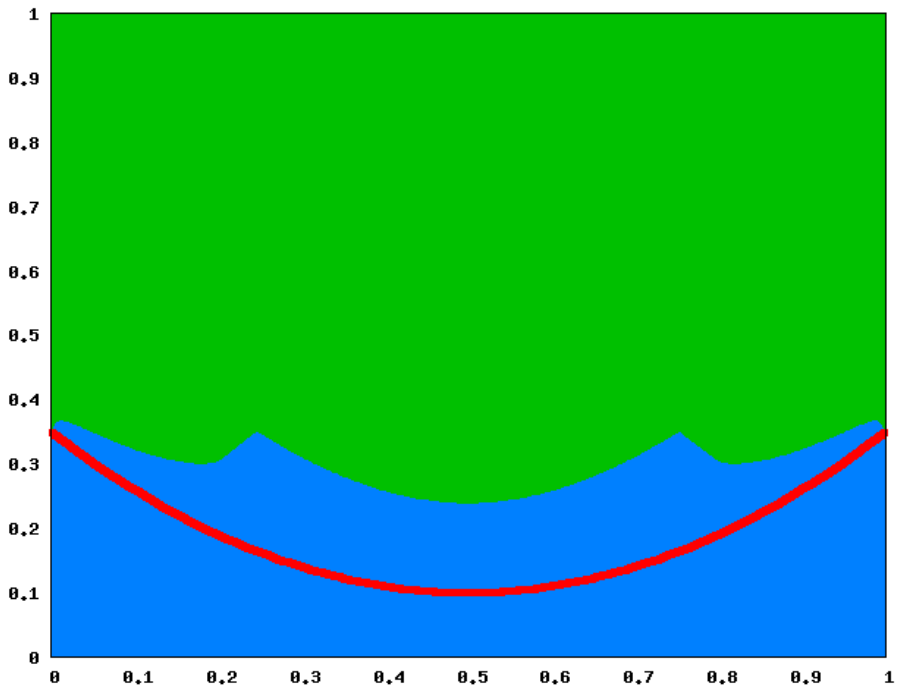
VI with 129×129 points grid

iterations $= 400$



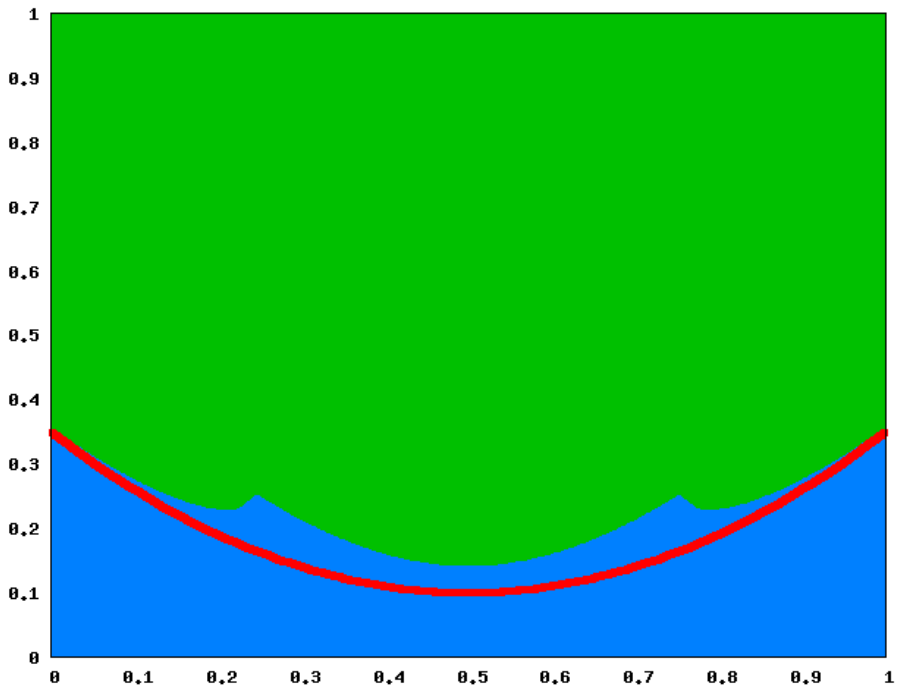
VI with 129×129 points grid

iterations $= 500$

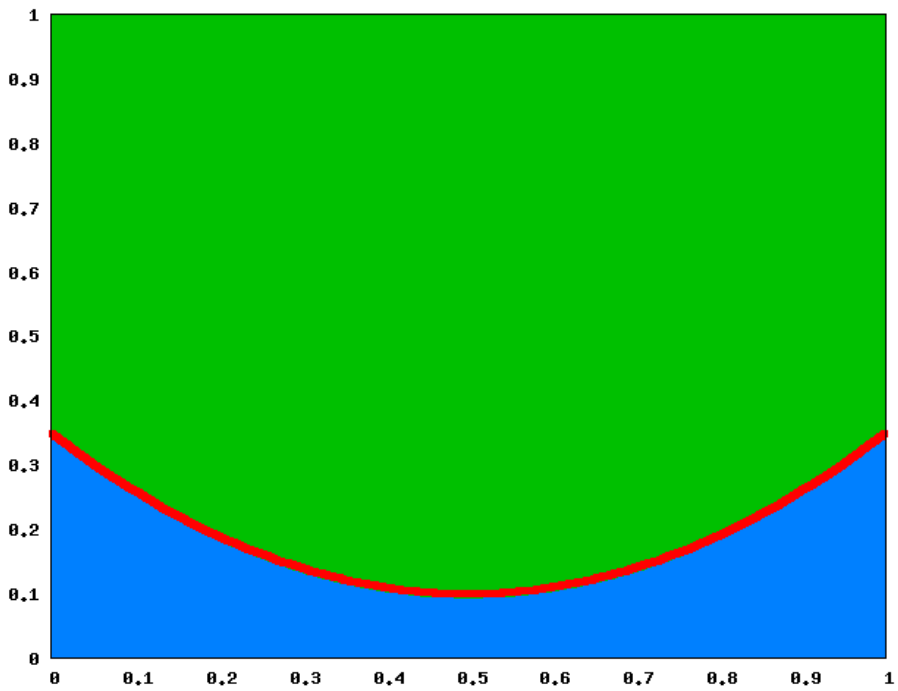


VI with 129×129 points grid

iterations $= 600$



VI with 129×129 points grid

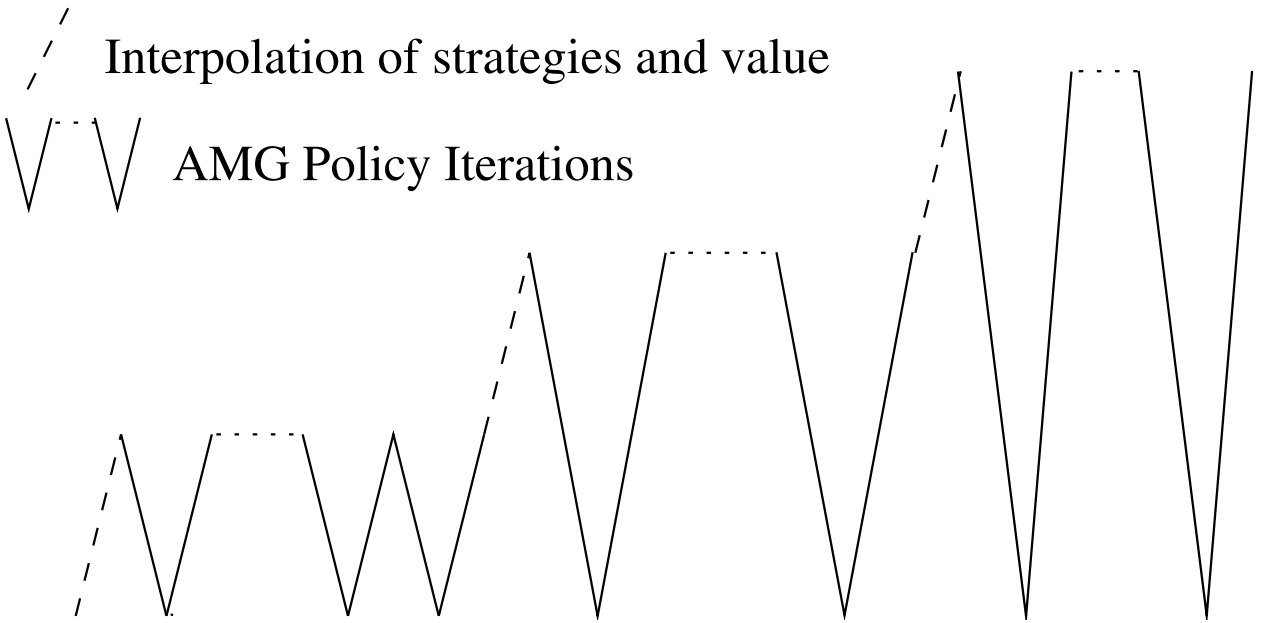


iteration 700!
in ≈ 8148 seconds
slow convergence
Policy iterations bounded by
$\sharp\{$possible policies$\}$
$=$ exponential in $\sharp\mathcal{X}$
[Friedmann, 09] example of parity game
[Fearnley, 10] for MDP

like Newton $\rightarrow$ improve with good initial guess? $\rightarrow$ *FMG*

Full Multilevel AMGπ

Define the problem on each coarse grid $\mathcal{X}_l := \{1, \cdots, n_l\}$ on level l



Interpolation of value v and strategies α, β
Stopping criterion for AMGπ $\|r\|_{L^2} < ch^2$ with $c = 0.1$ and $h = \frac{1}{n_l}$

Full Multilevel AMGπ

$\mathcal{X} = [0,1] \times [0,1]$, 1025 nodes in each direction
n_l = number of nodes in each direction (coarse grids)

n_l	MAX policy iteration index	Number of MIN policy iterations	$\|r\|_{L_2}$	$\|e\|_{L_2}$	CPU time (s)
3	1	1	$2.17e-1$	$1.53e-1$	$<<1$
3	2	1	$1.14e-2$	$3.30e-2$	$<<1$
5	1	2	$8.26e-5$	$1.71e-2$	$<<1$
9	1	2	$1.06e-3$	$7.99e-3$	$<<1$
9	2	1	$5.41e-4$	$8.15e-3$	$<<1$
9	3	1	$5.49e-5$	$8.30e-3$	$<<1$
$\vdots$					
513	1	1	$4.04e-9$	$1.33e-4$	2.62
1025	1	1	$1.90e-9$	$6.63e-5$	11.7
1025	2	1	$5.83e-10$	$6.62e-5$	21.1

Mean payoff of the game starting at $x \in \mathcal{X}$

$$\eta(x) = \sup_{(a_k)_{k\geq 0}} \inf_{(b_k)_{k\geq 0}} \limsup_{N\to\infty} \frac{1}{N} \mathbb{E}\left[\sum_{k=0}^{N} r(X_k, a_k, b_k)\right]$$

where

$$\begin{cases} a_k = a_k(X_k, b_{k-1}, a_{k-1}, X_{k-1} \cdots) \\ b_k = b_k(X_k, a_k, \cdots) \end{cases}$$

are strategies and the state dynamics satisfies the process X_k

$$P(X_{k+1} = y | X_k = x, a_k = a, b_k = b) = P(y|x, a, b)$$

Optimal strategies and dynamic programming

If there exist a constant $\rho \in \mathbb{R}$ and $v \in \mathbb{R}^n$ such that

$$\rho + v(x) = \max_{a \in \mathcal{A}(x)} \min_{b \in \mathcal{B}(x,a)} \sum_{y \in \mathcal{X}} P(y|x,a,b)v(y) + r(x,a,b), \quad \text{(DP)}$$

$x \in \mathcal{X}$. Then $\eta(x) = \rho$ for $x \in \mathcal{X}$ and v is called the relative value.

Moreover, α, β given by (DP) equations are optimal feedback strategies for both players.

For instance when matrices $P^{\alpha,\beta}$ are irreducible for all α and β.

Policy Iteration for games (Hoffman and Karp, 66)

$$\rho + v(x) = \max_{a \in \mathcal{A}(x)} \underbrace{\min_{b \in \mathcal{B}(x,a)} \sum_{y \in \mathcal{X}} P(y|x,a,b)v(y) + r(x,a,b)}_{F(v;x,a)}$$

Start with $\alpha_0 : x \mapsto \alpha_0(x)$

1. Calculate value and bias (ρ^{k+1}, v^{k+1}) for policy α_k solution of

$$\rho^{k+1} + v^{k+1}(x) = F(v^{k+1}; x, \alpha_k(x)) \qquad x \in \mathcal{X}$$

   Solved with PI for 1PG
2. Improve the policy α_{k+1} for v^{k+1}

$$PI_{ext} \begin{cases} \alpha_0 & PI_{int} \begin{cases} \beta_{0,0} \\ \vdots \\ \beta_{0,s} \end{cases} \\ \vdots \\ \alpha_k \end{cases}$$

At each intern iteration of PI: $\rho + v = Pv + r$
and P an irreducible markovian matrix (row-sums $= 1$) :

- using the stationary probability of an irreducible Markov Chain:

$$\pi^T P = \pi^T$$

$$\rho = \pi^T r \qquad v = Pv + r - \rho$$

$\rightarrow$ direct solver or linear solver

- by iterating on ρ and v alternatively

$$\rho = \nu(Pv + r - v)$$

$$v = Pv + r - \rho$$

$$\mu v = 0$$

with $\nu, \mu \in \mathbb{R}^n_+$ probability vectors $\rightarrow$ adapted AMG

Denote by $\mathbb{R}^{n\times n}_+ := \{A \in \mathbb{R}^{n\times n} \mid a_{ij} \geq 0, \text{ for } 1 \leq i,j \leq n\}$.

Theorem

Assume that $P \in \mathbb{R}^{n\times n}_+$ is an irreducible stochastic matrix. Let $A = I - P$ and decompose $A = M - N$ such that $M \in \mathbb{R}^{n\times n}_+$ is invertible and $S = M^{-1}N \in \mathbb{R}^{n\times n}_+$. Consider the iterates

$$\begin{aligned} v^{k+1} &= (I - 1\mu)(Sv^k + M^{-1}(r - \rho^k 1)), \\ \rho^{k+1} &= \nu(r - Av^{k+1}), \end{aligned}$$

where μ, ν are probability vectors. Then, the iterates converge to a solution if $\rho((I - 1\nu)NM^{-1}) < 1$.

Example on a pursuit-evasion game

Solve the stationary Isaacs equation on $\mathcal{X} = [-1/2, 1/2]^2$:

$$-\rho + \varepsilon \Delta v(x) + \max_{a \in \mathcal{A}} (a \cdot \nabla v(x)) + \min_{b \in \mathcal{B}} (b \cdot \nabla v(x)) + \|x\|_2^2 = 0$$

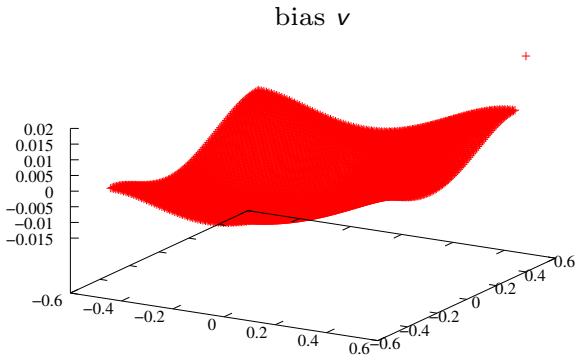
with $\epsilon = 0.5$ and Neumann boundary conditions.

$x = x_E - x_P$ with
x_E = position of evader (King)
x_P = position of pursuer (Horse)

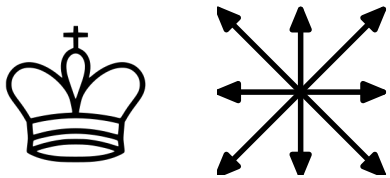
Actions for the King:
$\mathcal{A} := \{(a_1, a_2) \,|\, a_i = \pm 1 \text{ or } 0\}$
Actions for the Horse:
$\mathcal{B} := \{(0,0), (1,2), (2,1)\}$.

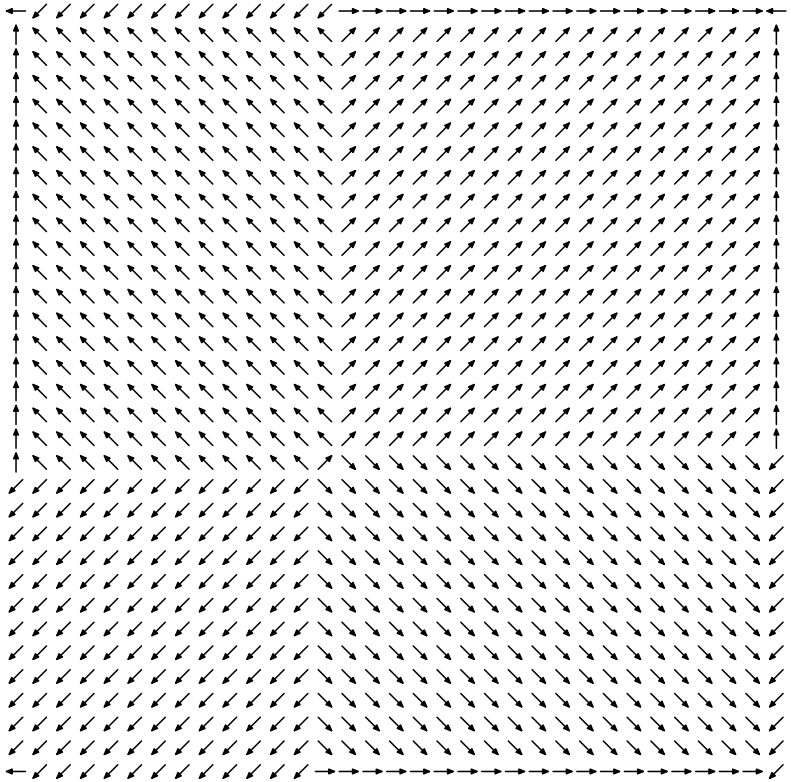
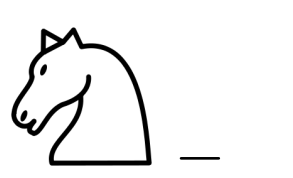
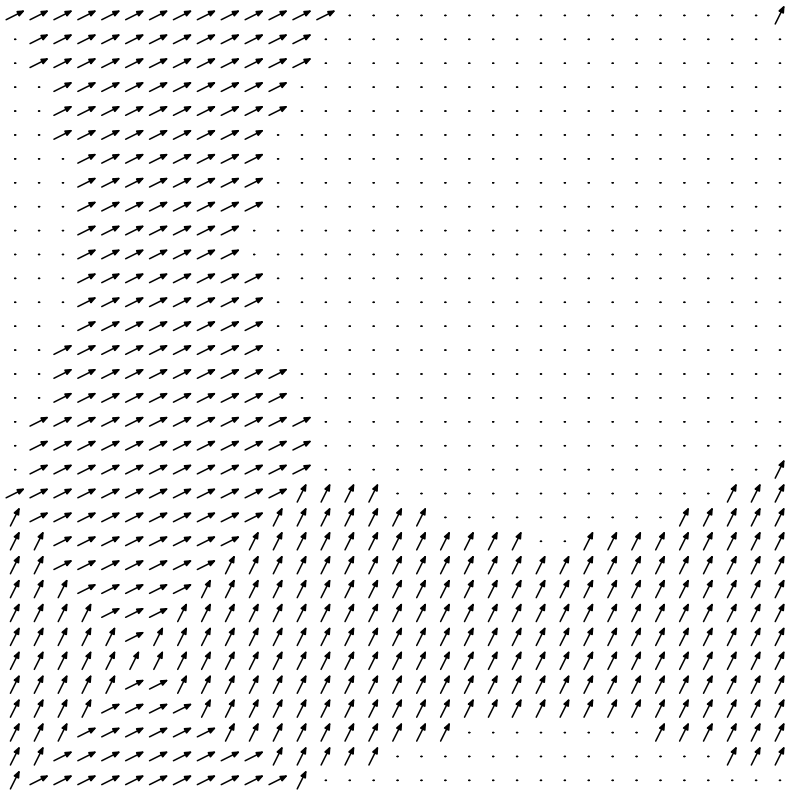
for a 129×129 grid :

$$\rho = 0,194$$



Optimal strategies



Numerical results

- PI & LU solver (SuperLU library using the stationary probability)

257x257 points grid

k	s	$\|r\|_\infty$	time
1	4	$4.54e-08$	$24s$
2	3	$5.87e-09$	$43s$
3	1	$6.97e-11$	$50s$

513x513 points grid

k	s	$\|r\|_\infty$	time
1	4	$2.27e-08$	$154s$
2	2	$3.27e-09$	$231s$
3	1	$4.78e-11$	$269s$

- PI & Adapted AMG (Ruge and Stuben algorithm computing ρ)

257x257 points grid

k	s	$\|r\|_\infty$	time
1	4	$4.54e-08$	$22s$
2	3	$5.87e-09$	$41s$
3	1	$6.97e-11$	$47s$

513x513 points grid

k	s	$\|r\|_\infty$	time
1	4	$2.27e-08$	$112s$
2	2	$3.27e-09$	$169s$
3	1	$4.78e-11$	$198s$

using $V(1,1)$-cycles (sym GS smoother), number of V-cycles ≈ 7

k = current iteration for MAX, s = number of iterations for MIN

Application: Perron eigenvector and eigenvalue

Assume $A \in \mathbb{R}_+^{n\times n}$ irreducible, the Perron eigenvector v and eigenvalue ρ is solution of $Av = \rho v \quad \rho > 0,\ v(i) > 0\, \forall i$

Set $v = \exp(w), w \in \mathbb{R}^n$, then we have to solve

$$\log \rho + w = F(w)$$

$$F_i(v) = \sup_{u\in\mathcal{A}_i} \left(u\, v - \sum_{\substack{j\in[n],\\ A_{ij}\neq 0}} \log\left(\frac{u_j}{A_{ij}}\right) u_j \right), \quad v \in \mathbb{R}^n, i \in [n]$$

where $\mathcal{A}_i = \left\{ u \in \mathbb{R}_+^n \mid u \text{ probability row-vector and } u \ll A_{i\cdot} \right\}$.

Apply to $A = P^T$ to find the stationary probability of an irreducible MC, we tested PI with adapted AMG versus MAA of [DeSterck, 08].

Policy Iteration for games (Hoffman and Karp, 66)

$$\rho + v(x) = \max_{a \in \mathcal{A}(x)} \underbrace{\min_{b \in \mathcal{B}(x,a)} \sum_{y \in \mathcal{X}} P(y|x,a,b)v(y) + r(x,a,b)}_{F(v;x,a)}$$

Start with $\alpha_0 : x \mapsto \alpha_0(x)$

1. Calculate value and bias (ρ^{k+1}, v^{k+1}) for policy α_k solution of

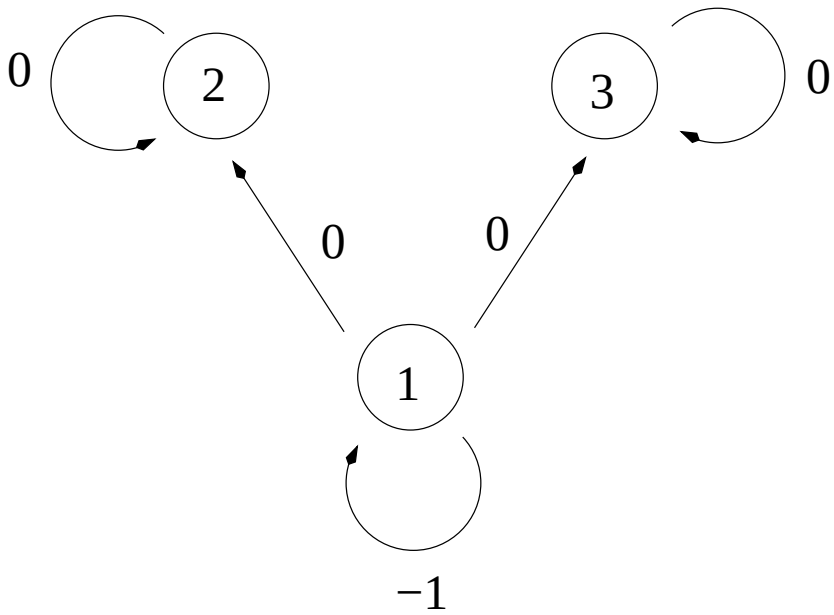
$$\rho^{k+1} + v^{k+1}(x) = F(v^{k+1}; x, \alpha_k(x)) \qquad x \in \mathcal{X}$$

   Solved with PI for 1PG
2. Improve the policy α_{k+1} for v^{k+1}

$$PI_{ext} \begin{cases} \alpha_0 & PI_{int} \begin{cases} \beta_{0,0} \\ \vdots \\ \beta_{0,s} \end{cases} \\ \vdots \\ \alpha_k \end{cases}$$

Variant of Richman game

$$f(v;x) = \frac{1}{2}\left(\max_{y:(x,y)\in E}(r(x,y)+v(y)) + \min_{y:(x,y)\in E}(r(x,y)+v(y))\right)$$



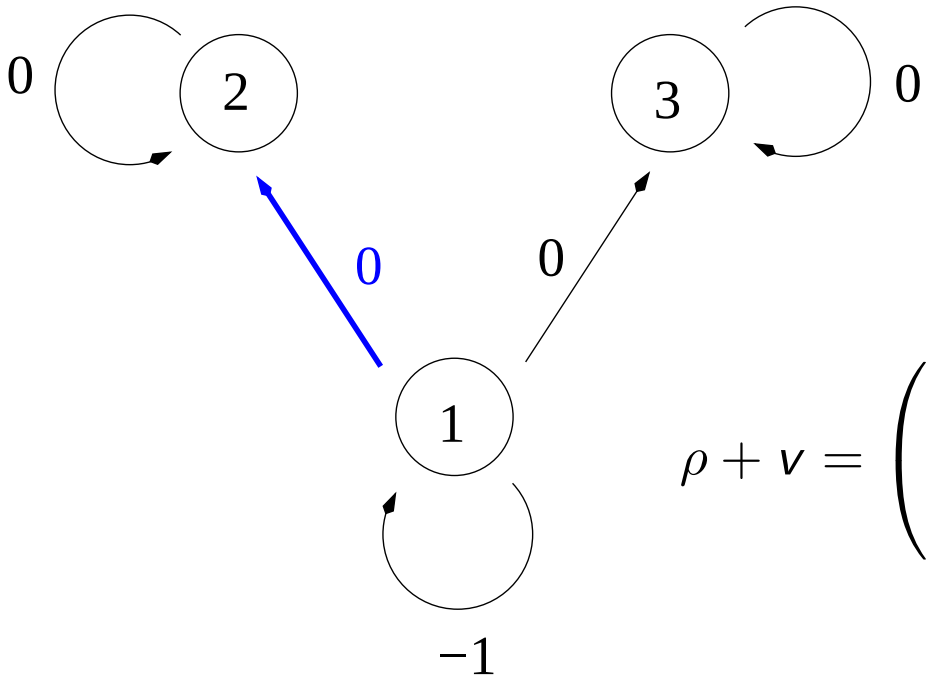
MAX and MIN flip a coin to decide who makes the move.

Min pays r to MAX.

$\mathcal{X} = \{1,2,3\}$
$E = \{(1,1),(1,2),(1,3),$
$(2,2),(3,3)\}$

Application of PI algorithm

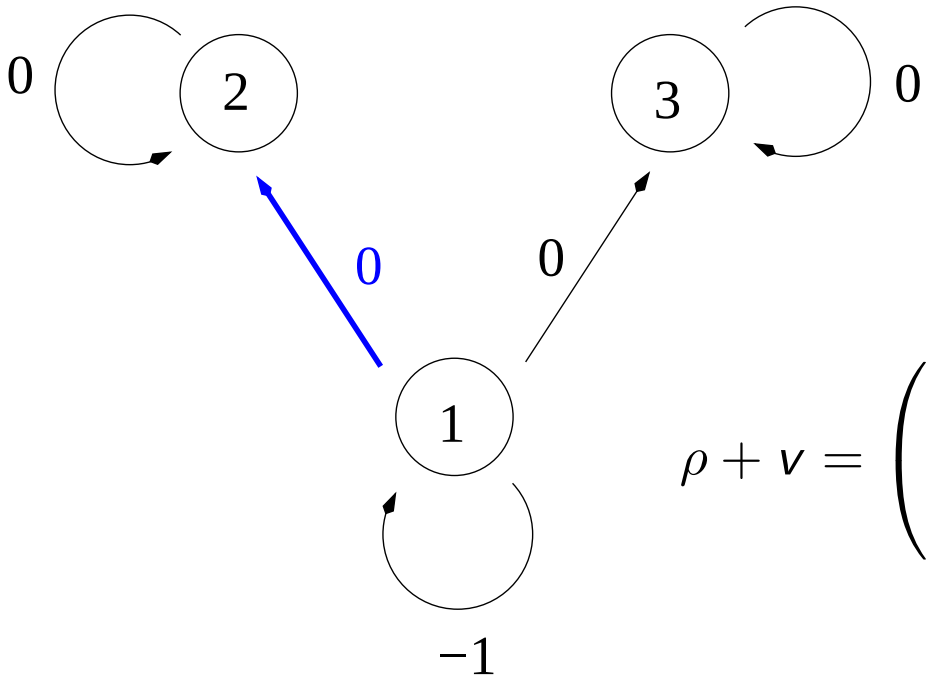
$$\rho + v = \begin{pmatrix} \frac{1}{2}\left(\max(v(1)-1, v(2), v(3)) + \min(v(1)-1, v(2), v(3))\right) \\ v(2) \\ v(3) \end{pmatrix}$$



$$\rho + v = \begin{pmatrix} \frac{1}{2}\left(\min(v(1)-1, v(2), v(3)) + v(2)\right) \\ v(2) \\ v(3) \end{pmatrix}$$

Application of PI algorithm

$$\rho + v = \begin{pmatrix} \frac{1}{2}\left(\max(v(1)-1, v(2), v(3)) + \min(v(1)-1, v(2), v(3))\right) \\ v(2) \\ v(3) \end{pmatrix}$$

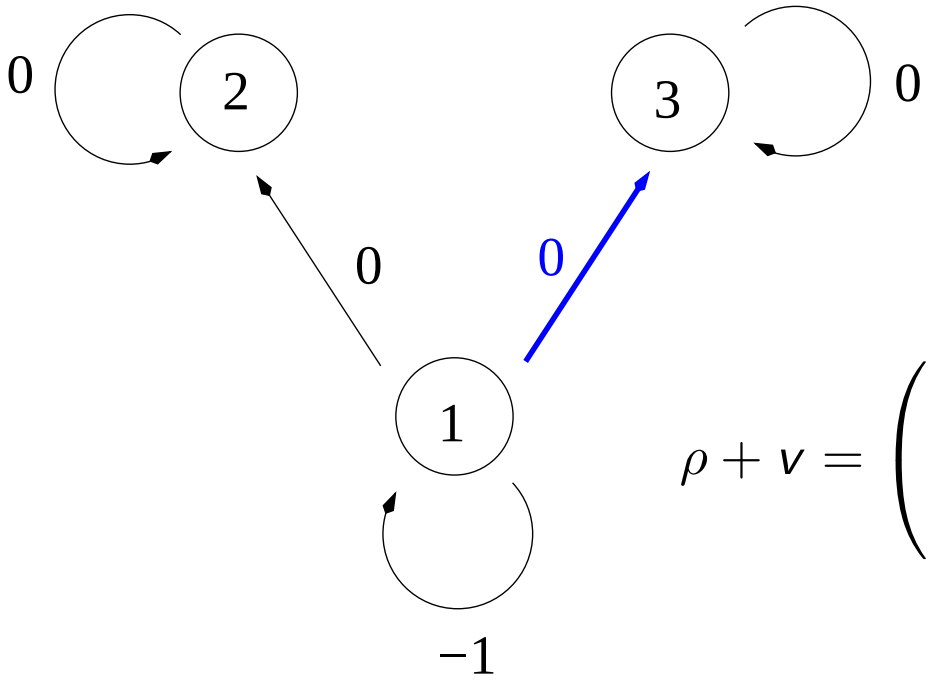


$$\rho + v = \begin{pmatrix} \frac{1}{2}\left(\min(v(1)-1, v(2), v(3)) + v(2)\right) \\ v(2) \\ v(3) \end{pmatrix}$$

$$v^{(1)} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \quad \rho = 0$$

Application of PI algorithm

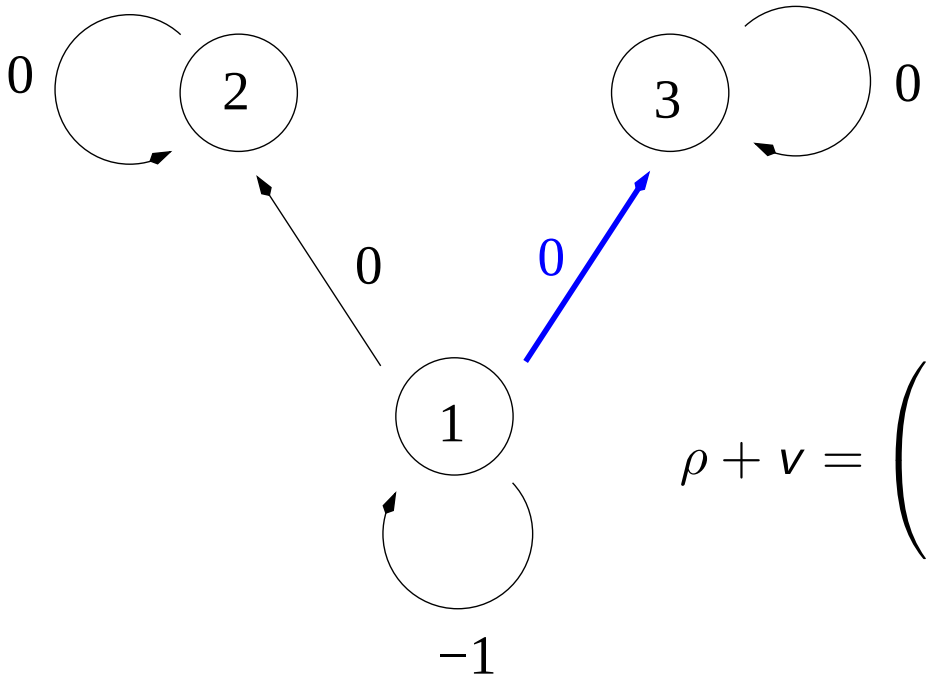
$$\rho + v = \begin{pmatrix} \frac{1}{2}\left(\max(v(1)-1, v(2), v(3) + \min(v(1)-1, v(2), v(3))\right) \\ v(2) \\ v(3) \end{pmatrix}$$



$$\rho + v = \begin{pmatrix} \frac{1}{2}\left(\min(v(1)-1, v(2), v(3)) + v(3)\right) \\ v(2) \\ v(3) \end{pmatrix}$$

Application of PI algorithm

$$\rho + v = \begin{pmatrix} \frac{1}{2}\left(\max(v(1)-1, v(2), v(3)) + \min(v(1)-1, v(2), v(3))\right) \\ v(2) \\ v(3) \end{pmatrix}$$

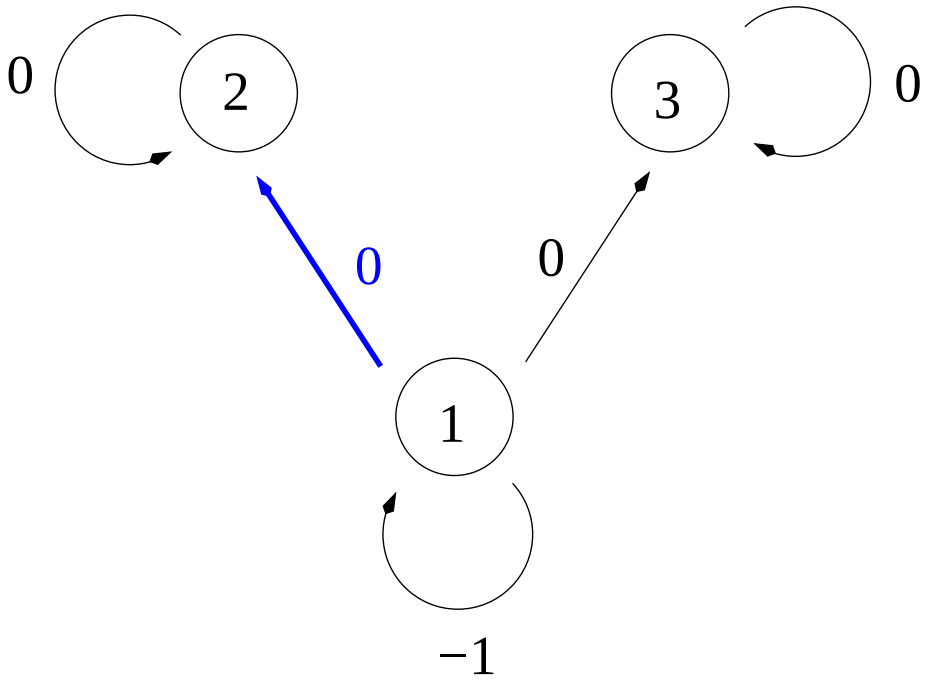


$$\rho + v = \begin{pmatrix} \frac{1}{2}\left(\min(v(1)-1, v(2), v(3)) + v(3)\right) \\ v(2) \\ v(3) \end{pmatrix}$$

$$v^{(2)} = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}, \quad \rho = 0$$

Application of PI algorithm

$$\rho + v = \begin{pmatrix} \frac{1}{2}\left(\max(v(1)-1, v(2), v(3)) + \min(v(1)-1, v(2), v(3))\right) \\ v(2) \\ v(3) \end{pmatrix}$$



$\alpha_3 = \alpha_1$

$\rightarrow$ Algorithm cycle!

Dynamic programming for multichain games

Assume $\mathcal{X} := \{1, \cdots, n\}$, $\mathcal{A}(x)$, $\mathcal{B}(x,a)$ are finite sets.
In general, the value η of the game is solution of the dynamic programming equation:

$$\eta(x)\,(t+1) + v(x) = F(\eta\, t + v; x), \quad x \in \mathcal{X},\ t \text{ large enough}$$

for some $v \in \mathbb{R}^n$, where F is the dynamic programming operator:

$$F(v;x) := \max_{a \in \mathcal{A}(x)} \min_{b \in \mathcal{B}(x,a)} \sum_{y \in \mathcal{X}} P(y|x,a,b)v(y) + r(x,a,b).$$

($\{\eta t + v, t \text{ large }\}$ is an invariant half line).

[Kolberg, 80]

This is equivalent to solve the system for $x \in \mathcal{X}$:

$$\begin{cases} \eta(x) = \max\limits_{a \in \mathcal{A}(x)} \min\limits_{b \in \mathcal{B}(x,a)} \sum\limits_{y \in \mathcal{X}} P(y|x,a,b)\ \eta(y) \\ \eta(x) + v(x) = \max\limits_{a \in \mathcal{A}_\eta(x)} \min\limits_{b \in \mathcal{B}_\eta(x,a)} \sum\limits_{y \in \mathcal{X}} P(y|x,a,b)\ v(y) \ + \ r(x,a,b) \end{cases}$$

with $\mathcal{A}_\eta(x) := \text{argmax}_{a \in \mathcal{A}(x)} \left\{ \min_{b \in \mathcal{B}(x,a)} \sum_{y \in \mathcal{X}} P(y|x,a,b)\ \eta(y) \right\}$
and $\mathcal{B}_\eta(x,a) := \text{argmin}_{b \in \mathcal{B}(x,a)} \left\{ \sum_{y \in \mathcal{X}} P(y|x,a,b)\ \eta(y) \right\}$.

DP for 1 player stochastic game with mean payoff

$$\begin{cases} \eta(x) = \min\limits_{b \in \mathcal{B}(x)} \sum\limits_{y \in \mathcal{X}} P(y|x,b)\, \eta(y) \\ \eta(x) + v(x) = \min\limits_{b \in \mathcal{B}_\eta(x)} \sum\limits_{y \in \mathcal{X}} P(y|x,b)\, v(y) \; + \; r(x,b) \end{cases}$$

where $x \in \mathcal{X}$ and $\mathcal{B}_\eta(x) = \text{argmin}_{b \in \mathcal{B}(x)} \left\{ \sum_{y \in \mathcal{X}} P(y|x,b)\, \eta(y) \right\}$.

Multichain Policy Iteration for 1PG (Howard, 60 and Denardo, Fox, 67)

Start with $\beta_0 : x \mapsto \beta_0(x)$, apply successively

1. Calculate value and bias (η^{s+1}, v^{s+1}) for policy β_s solution of

$$\eta^{s+1} = P^{\beta_s}\eta^{s+1} \quad \text{and} \quad \eta^{s+1} + v^{s+1} = P^{\beta_s}v^{s+1} + r^{\beta_s}$$

2. Improve the policy: select β_{s+1} optimal for (η^{s+1}, v^{s+1})

$$\beta_{s+1}(x) \in \operatorname*{argmin}_{b \in \mathcal{B}_{\eta^{s+1}}(x)} \left\{ \sum_{y \in \mathcal{X}} P(y|x,b)v^{s+1}(y) + r(x,b) \right\}$$

until $\beta_{s+1}(x) = \beta_s(x)\ \forall x \in \mathcal{X}$.

Degenerate iteration

Easy to show $\eta^{s+1} \leq \eta^s$

if $\eta^{s+1} = \eta^s \rightarrow$ degenerate iteration

v^{s+1} is defined up to $Ker(I - P^{\beta_s})$ with dim = nb of final class of P^{β_s}.

$\rightarrow$ PI may cycle when they are multiple final classes

To avoid this :

- Strategies are improved in a conservative way
  ($\beta_{s+1}(x) = \beta_s(x)$ if optimal)
- v^{s+1} is fixed on a point of each final class of P^{β_s}

$\Rightarrow$ when $\eta^{s+1} = \eta^s$, $v^{s+1}(x) = v^s(x)$ on each final classes of P^{β_s}

$\Rightarrow$ $(\eta^s, v^s)_{s \geq 1}$ is non increasing in a lexicographical order
$\eta^{s+1} \leq \eta^s$ and if $\eta^{s+1} = \eta^s$, $v^{s+1} \leq v^s$

$\Rightarrow$ PI stops after a finite time when sets of actions are finite

DP for 2 player stochastic game with mean payoff

$$\begin{cases} \eta(x) = \max\limits_{a \in \mathcal{A}(x)} \hat{F}(\eta; x, a) \\ \eta(x) + v(x) = \max\limits_{a \in \mathcal{A}_\eta(x)} \acute{F}_\eta(v; x, a) \end{cases}$$

where $x \in \mathcal{X}$ and :

$$\hat{F}(\eta; x, a) := \min_{b \in \mathcal{B}(x)} \sum_{y \in \mathcal{X}} P(y|x, a, b)\, \eta(y)$$

$$\acute{F}_\eta(v; x, a) := \min_{b \in \mathcal{B}_\eta(x,a)} \sum_{y \in \mathcal{X}} P(y|x, a, b)\, v(y) + r(x, a, b)$$

Multichain Policy Iteration for 2PG (Cochet-Terrasson, Gaubert, 06)

Start with $\alpha_0 : x \mapsto \alpha_0(x)$, apply successively

1. Calculate value and bias (η^{k+1}, v^{k+1}) for policy α_k solution of

$$\begin{cases} \eta(x) = \hat{F}(\eta; x, \alpha_k(x)) \\ \eta(x) + v(x) = \acute{F}_\eta(v; x, \alpha_k(x)) \end{cases}$$

   Use PI for 1P multichain game D& F

2. Improve the policy α_k in a conservative way.

until $\alpha_{k+1}(x) = \alpha_k(x)\ \forall x \in \mathcal{X}$.

Same as in D& F, if $\eta^{k+1} = \eta^k$, the set of solutions v^{k+1} may be of dim $> 1 \rightarrow$ PI may cycle

If $\eta^{k+1} = \eta^k$, then define

$$\bar{g}(v; x) := \acute{F}_{\eta^{k+1}}(v; x, \alpha_{k+1}(x)) - \eta^{k+1}(x)$$

the DP operator of a one player game.

Compute the the critical graph of $\bar{g}$ as defined in (Akian, Gaubert 2003) by using a v' such that $\bar{g}(v') = v'$, for instance take $v' = v^{k+1}$.

Solve

$$\begin{cases} v^{k+1}(x) & = & \bar{g}(v^{k+1}; x) & x \in N^{k+1} \\ v^{k+1}(x) & = & v^k(x) & x \in C^{k+1} \end{cases}$$

where $N^k := \mathcal{X} \setminus C^k$.

Theorem

$(\eta^k, v^k, C^k)_{k\geq 1} \nearrow$ *non decreasing in a "lexicographical order":*

$$\eta^k \leq \eta^{k+1} \text{ and if } \eta^k = \eta^{k+1},\ v^k \leq v^{k+1} \text{ and } C^k \supset C^{k+1}$$

PI stops after a finite time when sets of actions are finite

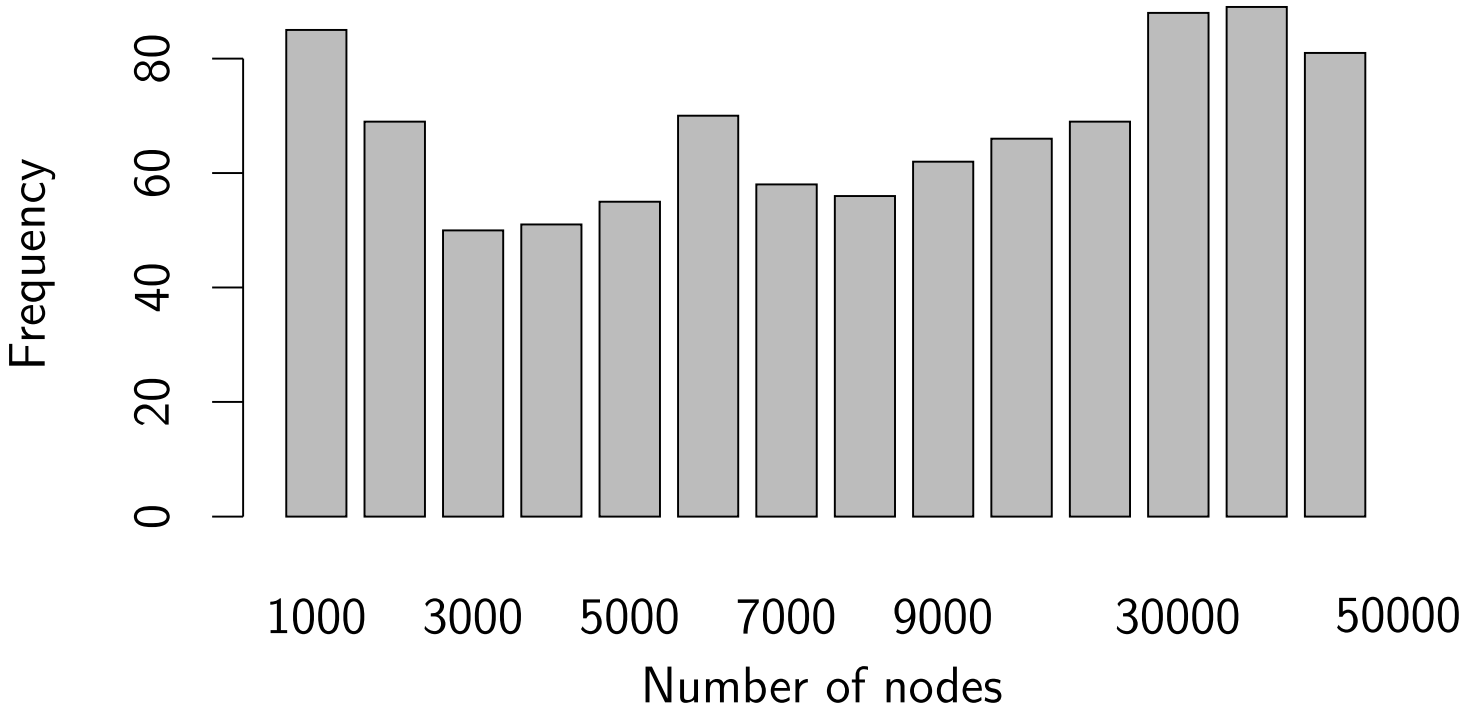
Solve Step 1 : $\eta = P\eta$ and $\eta + v = Pv + r$
Assume P has two final class and one transient class:

$$P = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ 0 & P_{22} & 0 \\ 0 & & P_{33} \end{pmatrix}$$

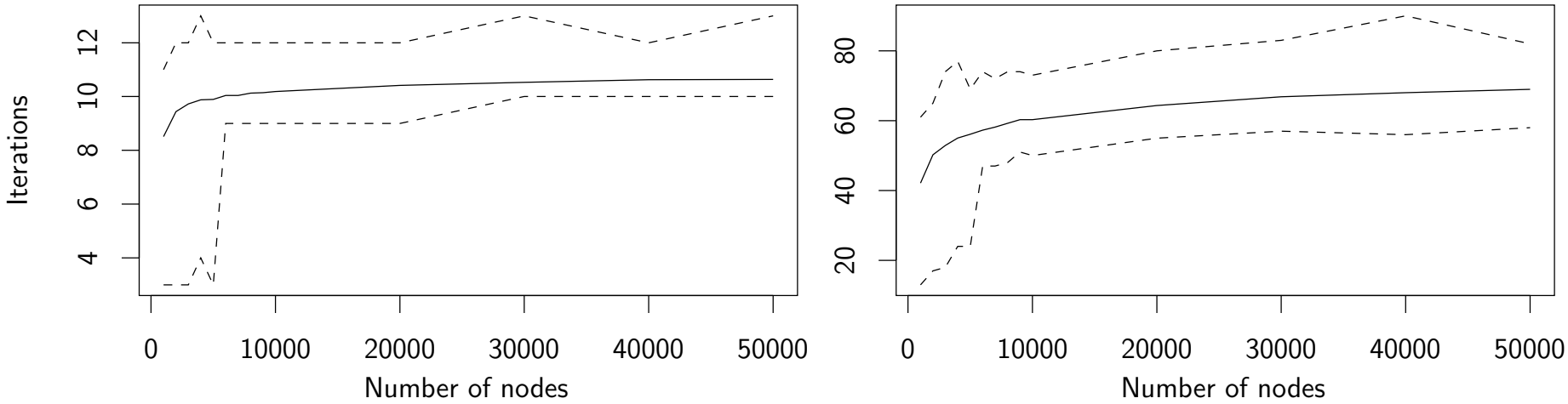
then we have to solve

1. For the final classes $I = 2, 3$:
   $\eta_I + v_I = P_{II} v_I + r_I, \qquad v_I(0) = 0, \eta_I(x) \equiv \eta_I, \ x \in I$
   with P_{II} an irreducible markovian matrix (row-sums $= 1$)
2. For the transient class 1:
   $\eta_1 = P_1\eta_1 + P_{12}\eta_2 + P_{13}\eta_3$
   $\eta_1 + v_1 = P_1 v_1 + P_{12} v_2 + P_{13} v_3 + r_1$
   with P_{11} an irreducible strictly submarkovian matrix
   (one row-sum < 1) $\rightarrow$ LU, AMG, etc

Richman game on random sparse graphs



10 arcs /node, 500 random graphs per dim, $>$ 10% strongly deg. iter.



Max, average, Min of policy iterations among 500 tests.

Left = extern PI (1st player)
Right = total intern PI (2nd player)

Instance for $n = 10^6$: 12 extern PI and 90 total intern PI

Example on a pursuit-evasion game

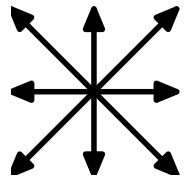
Set $x = x_E - x_P$ with $x_E =$ pos. of evader and $x_P =$ pos. of pursuer
Solve the stationary Isaacs equation on $\mathcal{X} = [-1/2, 1/2]^2$:

$$\begin{cases} \max_{a \in \mathcal{A}(x)} (a \cdot \nabla \eta(x)) + \min_{b \in \mathcal{B}(x)} (b \cdot \nabla \eta(x)) = 0 \ , & x \in \mathcal{X} \\ -\eta(x) + \max_{a \in \mathcal{A}_\eta(x)} (a \cdot \nabla v(x)) + \min_{b \in \mathcal{B}_\eta(x)} (b \cdot \nabla v(x)) + \|x\|_2^2 = 0 \ , & x \in \mathcal{X} \end{cases}$$

with natural boundary conditions (keeping x in the domain).

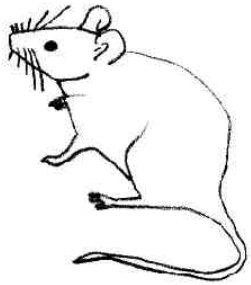
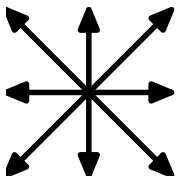
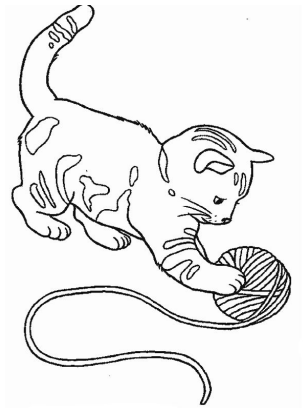
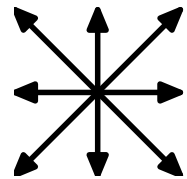
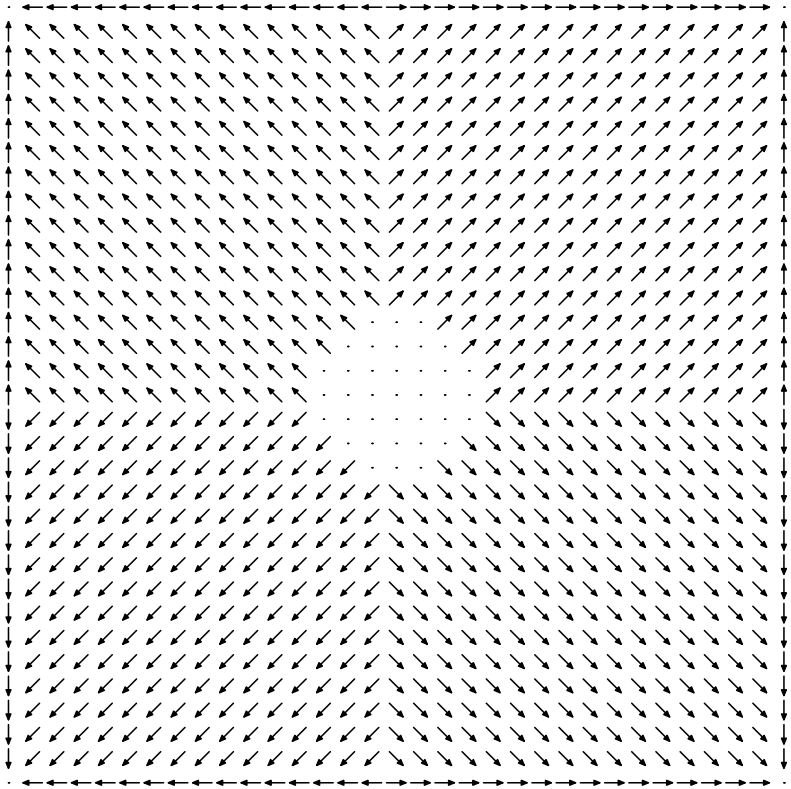
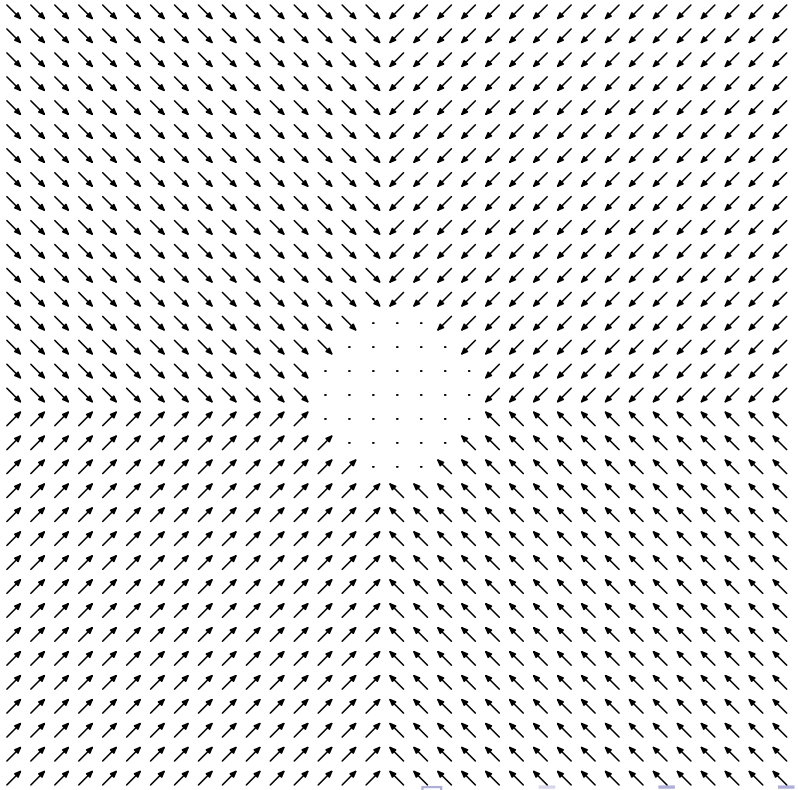
Actions for the Mouse:

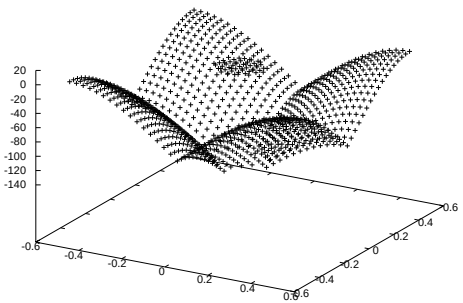
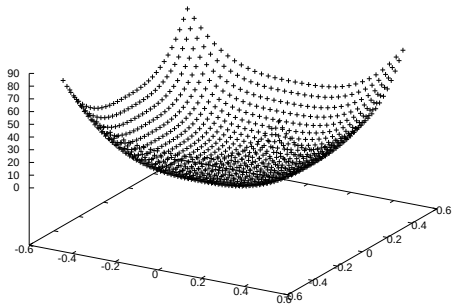
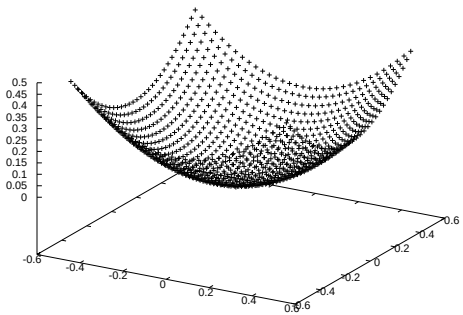
$$\mathcal{A}(x) := \begin{array}{ll} \{(0,0)\} & \text{if } x \in \mathcal{B}((0,0); 0.1) \\ \{(a_1, a_2) \,|\, a_i = \pm 1 \text{ or } 0\} & \text{otherwize} \end{array}$$



Actions for the Cat:

$\mathcal{B}(x) := \{(b_1, b_2) \,|\, b_i \in \{0, \bar{b}, -\bar{b}\}\}$, $\bar{b}$ constant

$\bar{b} = 0.999$	$\bar{b} = 1$	$\bar{b} = 1.001$
v	$v = 0$	v
		
$\eta = 0.492$	$\eta \approx \|x\|_2^2$	$\eta = 0$
		

$\bar{b}$	Cat policy iteration index	Number of mouse policy iterations	Infinite norm of residual	CPU time (s)
0.999	1	2	$1.25e-06$	$2.59e+01$
	2	1	$9.93e-12$	$3.95e+01$
	3	1	$5.68e-14$	$7.35e+02$
1	1	2	$1.25e-06$	$2.60e+01$
	2	1	$3.39e-21$	$3.84e+01$
1.001	1	2	$1.25e-06$	$2.59e+01$
	2	1	$1.96e-14$	$6.51e+02$

257 x 257 grid.

PIGAMES library

Implementation: PIGAMES (C library), by Detournay.

AMG, LU solver + decomposition into classes to solve linear systems. Double precision arithmetics.

In the double precision implementation, improvement tests are done up to some given treshold (which should be not too small if the matrices are ill conditioned).

Single proc. Intel(R) Xeon(R) W3540 - 2.93GHz with 8Go of RAM

Conclusions and Perspectives

- We have proposed algorithms combining AMG with PI for discounted stochastic games and unichain stochastic games with mean reward.
- AMG not efficient for strongly non symmetric matrices $->$ difficult to apply to general games
- Full multilevel scheme can make policy iteration faster and efficient!
- We have introduced a PI algorithm for multichain games and shown that degenerate iterations often occur.
- The termination proof of PI has been done assuming exact arithmetics.

Conclusions and Perspectives

- Find AMG for strongly unsymmetric systems to solve more general discrete games.
- Prove the convergence of a ϵ-approximate policy iteration algorithm.
- Estimation of the number of iterations as a function of the conditionning or the stationary probability of $P^{\alpha\beta}$?

- Akian, M. and Detournay, S. (2012), Multigrid methods for two-player zero-sum stochastic games. Numerical Linear Algebra with Applications.
- Akian M., Cochet-Terrasson J., Detournay S. and Gaubert S. (2012), Policy iteration algorithm for zero-sum multichain stochastic games with mean payoff and perfect information. Preprint on arXiv:1208.0446

Thank you!