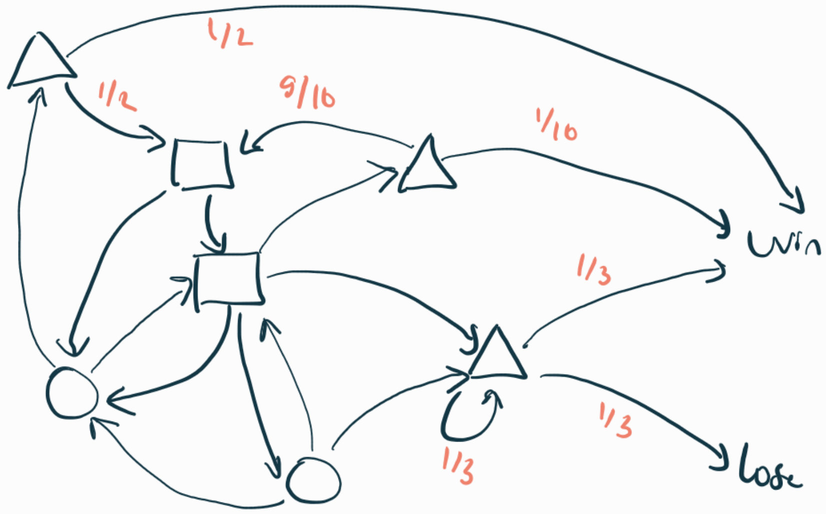
POSITIVE AND ALMOST-SURE

Step 1 : $W_{>0}$ positive

Step 2 : $W_{=1}$ almost sure



$$W_{>0} = LFP\left(X \mapsto Win \cup Pre_{>0}(X)\right)$$

$$W_{=1} = GFP\left(Y \mapsto W_{>0}(Win \cap Pre_{=1}(Y))\right)$$

$$\begin{cases} Y_0 = V \\ Y_{k+1} = W_{>0}\left(Win \cap Pre_{=1}(Y_k)\right) \end{cases}$$

or

$Attr_{>0}(Win) = V \Rightarrow V$

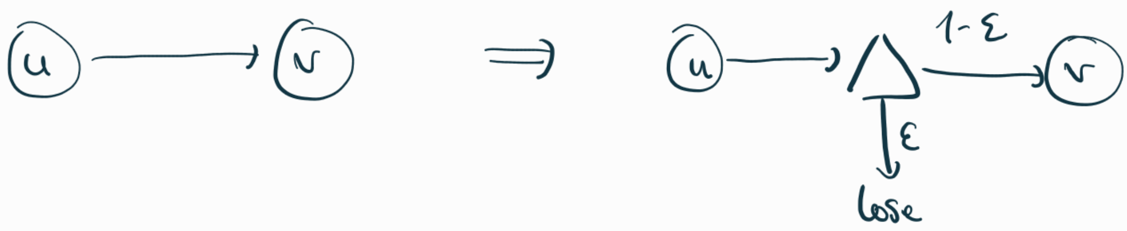
$Attr_{>0}(Win) \neq V \Rightarrow G' = G \mid Attr_{>0}(Win)$

$W_{=1}(G) = W_{=1}(G')$

STOPPING

Definition: $\forall \sigma, \tau, \forall v \quad \mathbb{P}_v^{\sigma,\tau}(\mathrm{Reach}(\mathrm{Win}, \mathrm{Lose})) = 1$

Reduction



for ε small enough:

- pure and positional optimal strategies coincide
- $val_G > \frac{1}{2} \iff val_{G'} > \frac{1}{2}$

Lemma: C absorbing Markov chain
where all probabilities are $\frac{x}{d}$ for $d \in \mathbb{N}_{>0}$

Then $\mathbb{P}(u \to v) = \frac{x'}{d^n}$ $\quad$ n #vertices

Fix $\varepsilon = \frac{1}{d^{2n}}$

$$\Phi(f)(v) = \begin{cases} 1 & v \in Win \\ 0 & v \in Lose \\ \max\{f(v') : v \to v' \in E\} & v \in V_{max} \\ \min\{f(v') : v \to v' \in E\} & v \in V_{min} \\ \sum_{v' \in V} \delta(v')(v)\, f(v') & v \in V_{Random} \end{cases}$$

Theorem: For stopping stochastic reachability games, $Val_{\mathcal{G}}$ is the unique fixed point of Φ

Lemma:

let f, define $f^* = \lim_n \Phi^n(f)$ it is well defined

monotonicity

- if $f \leq \Phi(f)$ then $f^* = LFP(\Phi)$
- if $f \geq \Phi(f)$ then $f^* = GFP(\Phi)$

LINEAR PROGRAMMING

computing the value of a stopping MDP :

$$\min \sum_{v \in V} val(v) \quad \text{such that}$$

$$\begin{cases} val(v) = 1 & \text{if } v \in Win \\ val(v) = 0 & \text{if } v \in Lose \\ val(v) \geq val(v') & \text{if } v \in V_{Max} \quad v \to v' \\ val(v) = \sum_{v' \in V} \delta(v' \mid v)\, val(v') & \end{cases}$$

what about stochastic game ?

abstract problem:

H finite set of constraints

$w: 2^H \longrightarrow \mathbb{R} \cup \{\pm\infty\}$

objective: find $B \subseteq H$ minimal
such that $w(B) = w(H)$

LP-type

(1) monotonicity $\quad F \subseteq G \subseteq H$

$$w(F) \leq w(G)$$

(2) locality $\quad F \subseteq G \subseteq H$

$$-\infty < w(F) = w(G)$$

$$\Longrightarrow \forall h \in H \quad \big(w(G \cup \{h\}) > w(G) \Rightarrow w(F \cup \{h\}) > w(F) \big)$$

G is feasible if $w(G) \neq \infty$

$B \subseteq H$ basis of G if $w(B) = w(G)$ and B minimal

combinatorial dimension:

$$\max \{ |\text{Basis of } G| \; : \; G \text{ feasible} \}$$

Remark: LP is an LP-type problem

$H = \{ (v,v') \in E : v \in V_{Max} \}$

$$w(S) = \begin{cases} -\infty & \text{if } \exists v \in V_{Max}\ \forall (v,v') \in E, (v,v') \notin E \\ \sum_{v \in V} \mathrm{val}_{\sigma}^{\mathcal{G}[S]}(v) \end{cases}$$

Recall $\mathrm{val}_{\sigma}(v) = \inf_{\tau} \mathbb{P}_{v}^{\sigma,\tau}(\mathrm{Reach}(\mathrm{Win}))$

Claims

(1) yields a reduction stochastic reachability games

(2) (H, w) is an LP-type problem

(3) LP-type problems can be solved in subexponential time

(1) monotonicity $S \subseteq S' \subseteq H$

$w(S) \leq w(S')$

S subgame $\Rightarrow$ S' subgame

(2) locality $S \subseteq S' \subseteq H$

$-\infty < w(S) = w(S')$

$\Rightarrow \forall e \in E \; \big(w(S' \cup \{e\}) > w(S') \Rightarrow w(S \cup \{e\}) > w(S) \big)$

(σ, τ) optimal in S $\Rightarrow$ (σ,τ) optimal in S'

Since $w(S' \cup \{e\}) > w(S')$: $e = (v,v')$ is switchable in S

so it is also switchable in S'

dimension: $|V_{Max}|$

SUBEXPONENTIAL TIME ALGORITHM

Two primitives:

(1) basis computation: | input G $|G| \leq d+1$
| outputs a basis of G

(2) violation test: | input B basis, $h \in H$
| outputs B basis for $B \cup \{h\}$?

- a basis for $G \subseteq H$ is an optimal strategy σ

  Given $|G| \leq d+1$ there are two strategies to compare

- B basis for $B \cup \{e = v \to v'\}$ iff

$$val_\sigma(v) \neq \max\{val_\sigma(\sigma(v)), val_\sigma(v')\}$$

Algorithm(G, C) — initial call: Algorithm(H, C) for $C \subseteq H$

If $G = C$ then output C

Else

- Choose $h \in G \setminus C$ at random
- $B \leftarrow$ Algorithm$(G \setminus \{h\}, C)$
- If Violation(B, h):
  - then Algorithm$(G$, Basis$(C \cup \{h\}))$
  - else Return B

$$e^{O(\sqrt{n \log n})}$$