Minimizing Regret in Infinite-Duration Games Played on Graphs

Guillermo A. Pérez
in collaboration with E. Filiot, P. Hunter,
I. Jecker, N. Lhote, J.-F. Raskin

Université libre de Bruxelles

LiVe 2018

What is this talk about?

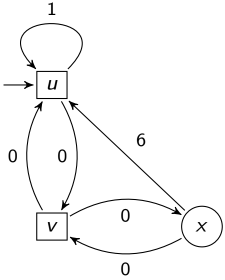
Key Words

Weighted graphs, ∃ve, ∀dam, Strategies, Infinite plays, and Payoff functions

What is this talk about?

Key Words

Weighted graphs, ∃ve, ∀dam, Strategies, Infinite plays, and Payoff functions



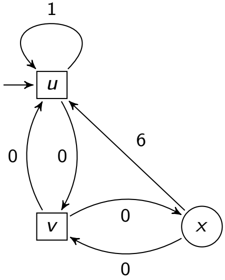
□ ∃ve

○ ∀dam

What is this talk about?

Key Words

Weighted graphs, ∃ve, ∀dam, Strategies, Infinite plays, and Payoff functions

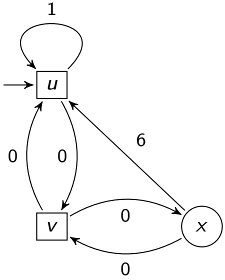


□	∃ve	$\sigma : V^* V_\exists \to V$
○	∀dam	$\tau : V^* V_\forall \to V$

What is this talk about?

Key Words

Weighted graphs, ∃ve, ∀dam, Strategies, Infinite plays, and Payoff functions



□	∃ve	$\sigma : V^* V_\exists \to V$
○	∀dam	$\tau : V^* V_\forall \to V$

$$\Big\} \quad \pi_{\sigma\tau} = v_0 v_1 \ldots$$

Which payoff functions?

A payoff function is of the form $\mathbf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$.

Which payoff functions?

A payoff function is of the form $\mathbf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$.

Classical payoff functions

- parity $\mathbb{N}^\omega \to \mathbb{B}$, positive energy $\mathbb{Z}^\omega \to \mathbb{B}$
- $\sup, \inf, \limsup, \liminf$, mean payoff, discounted sum

Which payoff functions?

A payoff function is of the form $\mathbf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$.

Classical payoff functions

- parity $\mathbb{N}^\omega \to \mathbb{B}$, positive energy $\mathbb{Z}^\omega \to \mathbb{B}$
- $\sup, \inf, \limsup, \liminf$, mean payoff, discounted sum

In this talk

We consider the mean-payoff function

$$\mathbf{Val}(x_1, x_2, \dots) = \liminf_{n \geq 1} \frac{1}{n} \sum_{i=1}^{n} x_i$$

Which payoff functions?

A payoff function is of the form $\mathbf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$.

Classical payoff functions

- parity $\mathbb{N}^\omega \to \mathbb{B}$, positive energy $\mathbb{Z}^\omega \to \mathbb{B}$
- $\sup, \inf, \limsup, \liminf,$ mean payoff, discounted sum

In this talk

We consider the mean-payoff function

$$\mathbf{Val}(x_1, x_2, \dots) = \liminf_{n \geq 1} \frac{1}{n} \sum_{i=1}^{n} x_i$$

Denote by $\mathbf{Val}(\sigma, \tau)$ the value $\mathbf{Val}(w(v_0, v_1), w(v_1, v_2), \dots)$ where $\pi_{\sigma\tau} = v_0 v_1 v_2 \dots$

Motivation 1: Modelling power

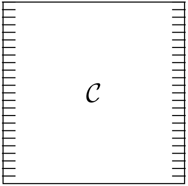
Mean-payoff games

are able to model [Zwick, Paterson 1996]

- online metrical task systems,
- finite window online string matching, and
- selection with limited storage.

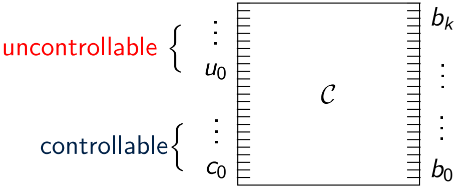
Motivation 2: Reactive synthesis

Consider a sequential circuit $\mathcal{C}$ with input set X partitioned into uncontrollable X_u and controllable X_c and output set B



Motivation 2: Reactive synthesis

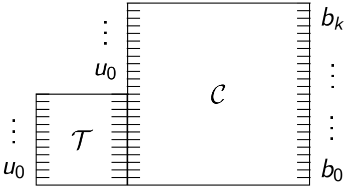
Consider a sequential circuit $\mathcal{C}$ with input set X partitioned into uncontrollable X_u and controllable X_c and output set B



$$\text{bin}(w_i) = (b_k, \ldots, b_0)$$

Motivation 2: Reactive synthesis

Consider a sequential circuit $\mathcal{C}$ with input set X partitioned into uncontrollable X_u and controllable X_c and output set B



$\text{bin}(w_i) = (b_k, \ldots, b_0)$

Synthesis

Does there exist $\mathcal{T}$ such that $\textbf{Val}(w_1, w_2, \ldots) \models \text{Spec}$ for all sequences of valuations of X_u?

Motivation 3: Interesting open problems

Existence of winning strategies

Does there exist a (finite memory) strategy σ of $\exists$ve such that

$$\inf_{\tau} \textbf{Val}(\sigma, \tau) \geq \ell?$$

Motivation 3: Interesting open problems

Existence of winning strategies

Does there exist a (finite memory) strategy σ of ∃ve such that

$$\inf_{\tau} \mathbf{Val}(\sigma, \tau) \geq \ell?$$

Parity games

Given a parity game, to determine if ∃ve has a winning strategy is in UP ∩ coUP [Jurdziński 1998] as well as in QP [Calude et al. 2017].

Motivation 3: Interesting open problems

Existence of winning strategies

Does there exist a (finite memory) strategy σ of ∃ve such that

$$\inf_{\tau} \mathbf{Val}(\sigma, \tau) \geq \ell?$$

Parity games

Given a parity game, to determine if ∃ve has a winning strategy is in $\mathrm{UP} \cap \mathrm{coUP}$ [Jurdziński 1998] as well as in QP [Calude et al. 2017].

Known reductions [Zwick, Paterson 1996; Jurdziński 1998]

The following hold when graphs are given explicitly; weights (and discount factor), in binary.

$$\textit{PGs} \leq_P \textit{MPGs} \leq_P \textit{DSGs} \leq_P \textit{SSGs}$$

What do you mean by regret minimization?

In words...

We want to find a strategy of $\exists$ve that minimizes the difference between her actual payoff and the payoff she could have achieved if she had known the strategy of $\forall$dam in advance.

What do you mean by regret minimization?

In words...

We want to find a strategy of ∃ve that minimizes the difference between her actual payoff and the payoff she could have achieved if she had known the strategy of ∀dam in advance.

Is it useful?

What do you mean by regret minimization?

In words...

We want to find a strategy of ∃ve that minimizes the difference between her actual payoff and the payoff she could have achieved if she had known the strategy of ∀dam in advance.

Is it useful?

- "A better solution concept than NE" [Halpern, Pass 2011]
  - less pessimistic than winning strategies

What do you mean by regret minimization?

In words...

We want to find a strategy of ∃ve that minimizes the difference between her actual payoff and the payoff she could have achieved if she had known the strategy of ∀dam in advance.

Is it useful?

- "A better solution concept than NE" [Halpern, Pass 2011]
  - less pessimistic than winning strategies
- Competitive analysis of
  - online metrical task systems
  - finite window online string matching
  - selection with limited storage

What do you mean by regret minimization?

In words...

We want to find a strategy of $\exists$ve that minimizes the difference between her actual payoff and the payoff she could have achieved if she had known the strategy of $\forall$dam in advance.

Is it useful?

- "A better solution concept than NE" [Halpern, Pass 2011]
  - less pessimistic than winning strategies
- Competitive analysis of
  - online metrical task systems
  - finite window online string matching
  - selection with limited storage
- Automata determinization by pruning [Aminof, Kupferman, Lampert 2010], good-for-games automata [Henzinger, Piterman 2006]

Values of a game

Let $\mathcal{G}$ be a game.

Cooperative value

$$\mathbf{cVal}(\mathcal{G}) := \sup_{\sigma} \sup_{\tau} \mathbf{Val}(\sigma, \tau)$$

Values of a game

Let $\mathcal{G}$ be a game.

Cooperative value

$$\mathbf{cVal}(\mathcal{G}) := \sup_{\sigma} \sup_{\tau} \mathbf{Val}(\sigma, \tau)$$

Antagonistic value

$$\mathbf{aVal}(\mathcal{G}) := \sup_{\sigma} \inf_{\tau} \mathbf{Val}(\sigma, \tau)$$

A formal definition of regret

Let $\Sigma_\exists$ be a set of strategies of $\exists$ve and $\Sigma_\forall$ a set of strategies of $\forall$dam.

The regret of σ

$$\mathbf{reg}^{\sigma}_{\Sigma_\exists,\Sigma_\forall}(\mathcal{G}) := \sup_{\tau \in \Sigma_\forall} \left(\overbrace{\sup_{\sigma' \in \Sigma_\exists} \mathbf{Val}(\sigma', \tau)}^{\text{best-response value}} - \mathbf{Val}(\sigma, \tau) \right)$$

A formal definition of regret

Let $\Sigma_\exists$ be a set of strategies of $\exists$ve and $\Sigma_\forall$ a set of strategies of $\forall$dam.

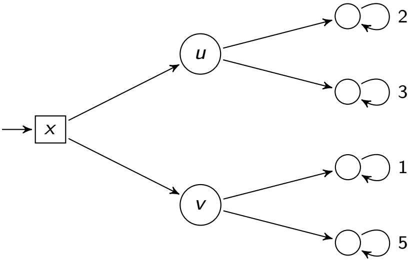
The regret of σ

$$\mathbf{reg}^{\sigma}_{\Sigma_\exists,\Sigma_\forall}(\mathcal{G}) := \sup_{\tau \in \Sigma_\forall} \left(\overbrace{\sup_{\sigma' \in \Sigma_\exists} \mathbf{Val}(\sigma', \tau)}^{\text{best-response value}} - \mathbf{Val}(\sigma, \tau) \right)$$

The regret of $\exists$ve in $\mathcal{G}$

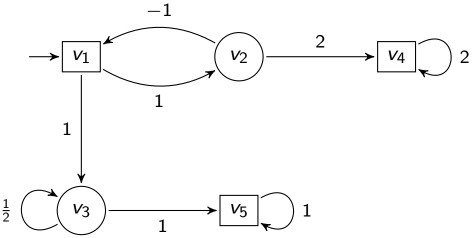
$$\mathbf{Reg}_{\Sigma_\exists,\Sigma_\forall}(\mathcal{G}) := \inf_{\sigma \in \Sigma_\exists} \mathbf{reg}^{\sigma}_{\Sigma_\exists,\Sigma_\forall}(\mathcal{G})$$

Example: simple tree-like mean-payoff game (MPG)



$\mathbf{cVal}(\mathcal{G}) = 5$, $\mathbf{aVal}(\mathcal{G}) = 2$, $\mathbf{Reg}(\mathcal{G}) = 2$

Example: it's a trap!



$\mathbf{cVal}(\mathcal{G}) = 2$, $\mathbf{aVal}(\mathcal{G}) = \frac{1}{2}$, $\mathbf{Reg}(\mathcal{G}) = 1$

Results

Theorem (Hardness)

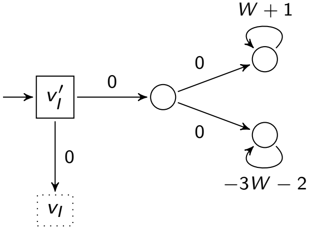
Computing the regret of a game is at least as hard as computing the antagonistic value of a (polynomial-size) game with the same payoff function.

From MPGs to MP Regret games

If $W := \max_{e \in E} |w(e)|$ then $-W \leq \mathbf{aVal}(\mathcal{G}) \leq \mathbf{cVal}(\mathcal{G}) \leq W$

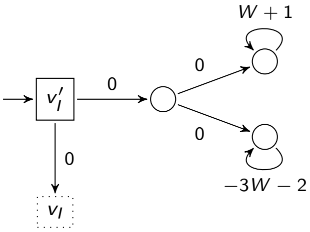
From MPGs to MP Regret games

If $W := \max_{e \in E} |w(e)|$ then $-W \leq \mathbf{aVal}(\mathcal{G}) \leq \mathbf{cVal}(\mathcal{G}) \leq W$



From MPGs to MP Regret games

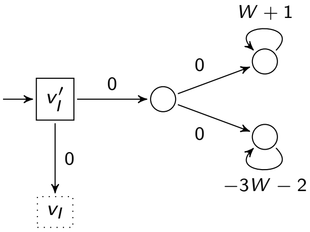
If $W := \max_{e \in E} |w(e)|$ then $-W \leq \mathbf{aVal}(\mathcal{G}) \leq \mathbf{cVal}(\mathcal{G}) \leq W$



1. ∃ve plays to v_I

From MPGs to MP Regret games

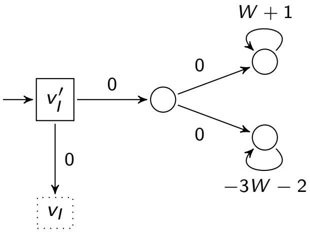
If $W := \max_{e \in E} |w(e)|$ then $-W \leq \mathbf{aVal}(\mathcal{G}) \leq \mathbf{cVal}(\mathcal{G}) \leq W$



1. $\exists$ve plays to v_I
2. $\forall$dam plays antagonistically from v_I and allows $W+1$ from v_I'

From MPGs to MP Regret games

If $W := \max_{e \in E} |w(e)|$ then $-W \leq \mathbf{aVal}(\mathcal{G}) \leq \mathbf{cVal}(\mathcal{G}) \leq W$



1. ∃ve plays to v_I
2. ∀dam plays antagonistically from v_I and allows $W+1$ from v'_I
3. the regret of ∃ve in the game is $W + 1 - \mathbf{aVal}(\mathcal{G})$

Results

Theorem (Hardness)

Computing the regret of a game is at least as hard as computing the antagonistic value of a (polynomial-size) game with the same payoff function.

Theorem (Algorithm)

Computing the regret reduces to computing the antagonistic value of a (polynomial-size) game with the same payoff function.

From MP Regret games to MPGs

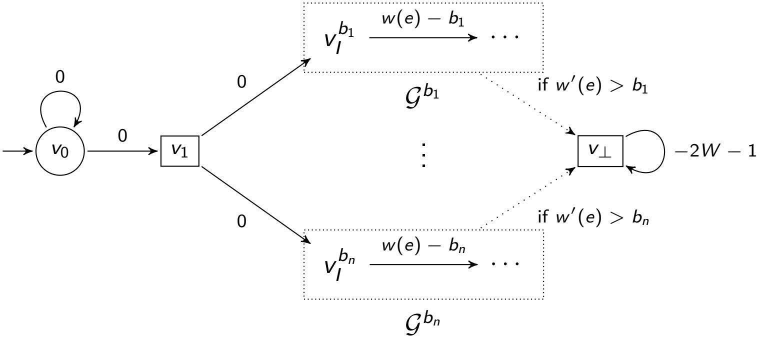
1. Label ∀dam edges as $w'(e) = -\infty$ and ∃ve edges as follows: $w'(e) = \max\{\mathbf{cVal}^{v'} : (u, v') \in E \setminus \{e\}\}$.

From MP Regret games to MPGs

1. Label ∀dam edges as $w'(e) = -\infty$ and ∃ve edges as follows: $w'(e) = \max\{\mathbf{cVal}^{v'} : (u, v') \in E \setminus \{e\}\}$.
2. Let $\mathcal{G}^b$ be the restriction of $\mathcal{G}$ to edges e with $w'(e) \leq b$.

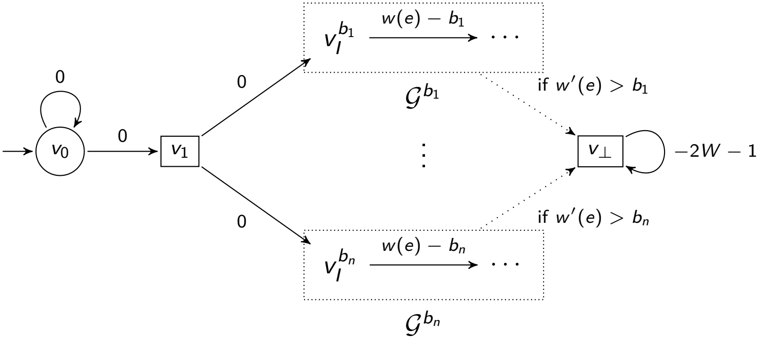
From MP Regret games to MPGs

1. Label ∀dam edges as $w'(e) = -\infty$ and ∃ve edges as follows: $w'(e) = \max\{\mathbf{cVal}^{v'} : (u, v') \in E \setminus \{e\}\}$.
2. Let $\mathcal{G}^b$ be the restriction of $\mathcal{G}$ to edges e with $w'(e) \leq b$.



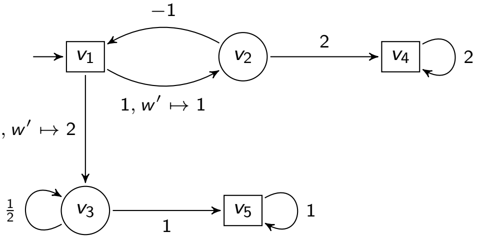
From MP Regret games to MPGs

1. Label ∀dam edges as $w'(e) = -\infty$ and ∃ve edges as follows: $w'(e) = \max\{\mathbf{cVal}^{v'} : (u, v') \in E \setminus \{e\}\}$.
2. Let $\mathcal{G}^b$ be the restriction of $\mathcal{G}$ to edges e with $w'(e) \leq b$.



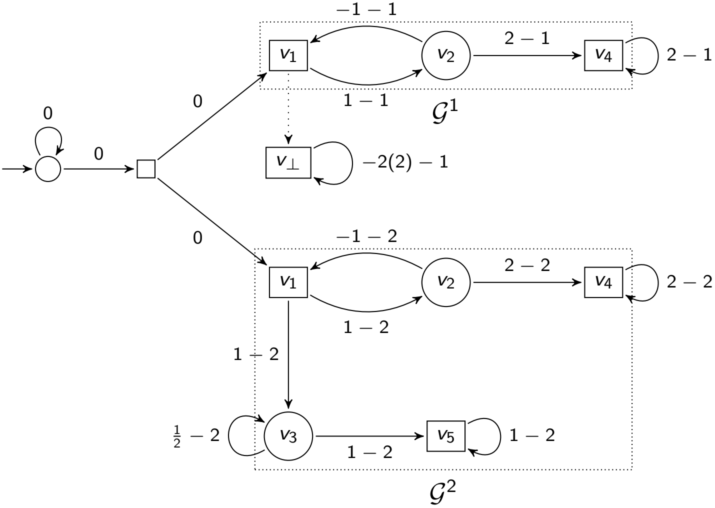
$\implies \mathbf{aVal}(\mathcal{G}') = -\mathbf{Reg}(\mathcal{G})$

Example: it's a trap!



$\mathbf{cVal}(\mathcal{G}) = 2$, $\mathbf{aVal}(\mathcal{G}) = \frac{1}{2}$, $\mathbf{Reg}(\mathcal{G}) = 1$

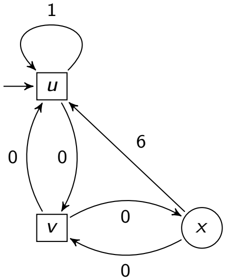
From MP Regret games to MPGs (example)



$\mathbf{aVal}(\mathcal{G}') = -1$

Motivating example: Learning in an MPG

Assume ∀dam plays positionally.



$\mathbf{cVal}(\mathcal{G}) = 2$, $\mathbf{aVal}(\mathcal{G}) = 1$, $\mathbf{Reg}_{\mathrm{Pos}_\forall}(\mathcal{G}) = 0$

Results

Theorem (Hardness)

Given $r \in \mathbb{Q}$ and a weighted graph $\mathcal{G}$, determining whether the regret value is less than r is PSPACE*-hard.*

Results

Theorem (Hardness)

Given $r \in \mathbb{Q}$ and a weighted graph $\mathcal{G}$, determining whether the regret value is less than r is PSPACE*-hard.*

Theorem (Algorithm)

The regret value can be computed using only polynomial space.

From MP Regret games to MPGs

We construct a new graph $\hat{\mathcal{G}}$ where

- the vertices record the witnessed choices of $\forall$dam

$$\hat{V} = V \times \mathcal{P}(E)$$

From MP Regret games to MPGs

We construct a new graph $\hat{\mathcal{G}}$ where

- the vertices record the witnessed choices of ∀dam

$$\hat{V} = V \times \mathcal{P}(E)$$

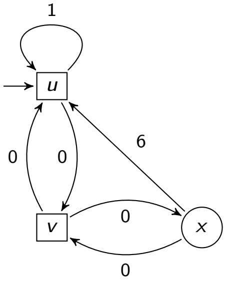
- the new weight function uses this info to reduce the value of potential alternatives

$$\hat{w}\big((u, C), (v, D)\big) = w(u, v) - \mathbf{cVal}(\mathcal{G} \cap D)$$

- $\mathbf{aVal}(\hat{\mathcal{G}}) = -\mathbf{Reg}_{\mathrm{Pos}_\forall}(\mathcal{G})$

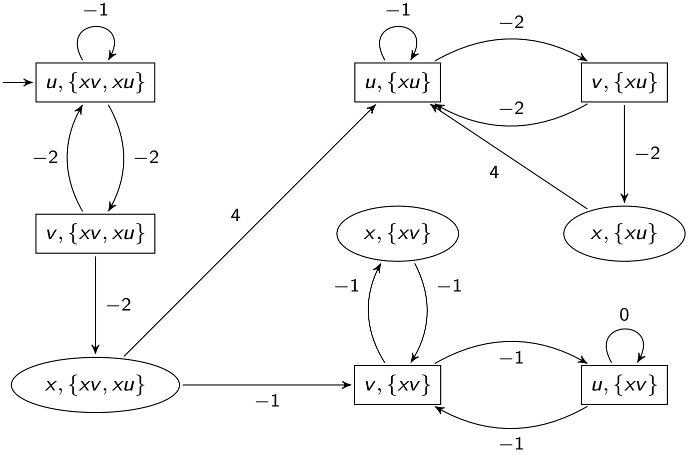
Motivating example: Learning in an MPG

Assume ∀dam plays positionally.



$\textbf{cVal}(\mathcal{G}) = 2$, $\textbf{aVal}(\mathcal{G}) = 1$, $\textbf{Reg}_{\mathrm{Pos}_\forall}(\mathcal{G}) = 0$

From MP Regret games to MPGs (example)



$\mathbf{aVal}(\hat{\mathcal{G}}) = 0$

Any questions?

	sup	inf	lim sup	lim inf	MP	DS
Any	poly-time equiv to regular game					in NP
Positional	$\in$ PSPACE			PSPACE-c		in EXP

Future work

- Compare to work on metrical task systems, etc.
- Improve SOTA for DS-games
- Compare to ML works on regret minimization

References

- On Delay and Regret Determinization of Max-Plus Automata. Filiot, Jecker, Lhote, P., Raskin. LICS 2017.
- Reactive Synthesis Without Regret. Acta Informatica 2017. Hunter, P., Raskin.
- Minimizing Regret in Discounted-Sum Games. CSL 2016. Hunter, P., Raskin.