Minimizing Regret in Discounted-Sum Games

Guillermo A. Pérez

joint work with Jean-François Raskin & Paul Hunter

CSL'16 @ Marseille

Minimizing Regret

Our Goal

We want to find a strategy that minimizes the difference between our payoff and the payoff we could have achieved if we had known the strategy of the adversary in advance.

Minimizing Regret

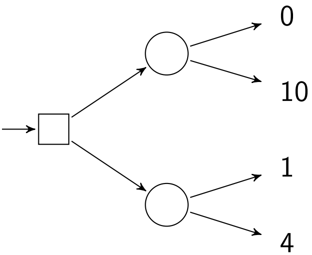
Our Goal

We want to find a strategy that minimizes the difference between our payoff and the payoff we could have achieved if we had known the strategy of the adversary in advance.

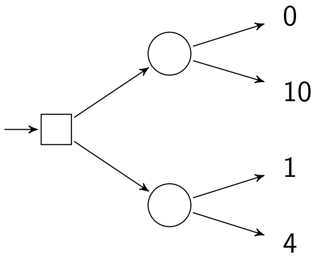
Halpern and Pass argue. . .

"A better solution concept than NE" because it captures "more rational" choices in classical games.

A first example

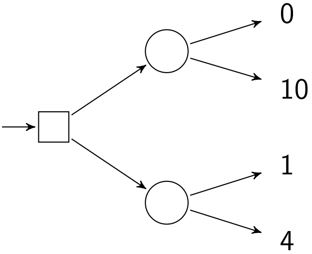


A first example



We can make sure our regret is at most 4.

A first example



We can make sure our regret is at most 4.
(In general, incomparable with worst-case optimal strategies)

Regret, formally (1/3)

Who is playing? What are their goals?

Regret, formally (1/3)

Who is playing? What are their goals?

Players

$\exists$ve will attempt to minimize her regret, while $\forall$dam tries to achieve the opposite.

Regret, formally (1/3)

Who is playing? What are their goals?

Players

$\exists$ve will attempt to minimize her regret, while $\forall$dam tries to achieve the opposite.

What do we play on?

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Regret, formally (1/3)

Who is playing? What are their goals?

Players

$\exists$ve will attempt to minimize her regret, while $\forall$dam tries to achieve the opposite.

What do we play on?

Weighted Graph

$G = (V, V_\exists, v_I, E, w)$ where (V, E) is a digraph, $V_\exists$ are the vertices of $\exists$ve (squares), v_I is the initial vertex, and $w : E \to \mathbb{Q}$ is a weight function.[a]

[a] Let W denote the "biggest" weight.

UNIVERSITÉ LIBRE DE BRUXELLES

Regret, formally (1/3)

Who is playing? What are their goals?

Players

∃ve will attempt to minimize her regret, while ∀dam tries to achieve the opposite.

What do we play on?

Weighted Graph

$G = (V, V_\exists, v_I, E, w)$ where (V, E) is a digraph, $V_\exists$ are the vertices of ∃ve (squares), v_I is the initial vertex, and $w : E \to \mathbb{Q}$ is a weight function.[a]

[a] Let W denote the "biggest" weight.

We consider infinite plays.

UNIVERSITÉ LIBRE DE BRUXELLES

Regret, formally (2/3)

What is the payoff/value of an infinite play?

Regret, formally (2/3)

What is the payoff/value of an infinite play?

Discounted-Sum

We aggregate the infinite weight sequence $\chi = x_0 x_1 \ldots$ with: $\sum_{i=0}^{\infty} \lambda^i x_i$, where $0 < \lambda < 1$ is a rational given as part of the input.

UNIVERSITÉ LIBRE DE BRUXELLES

Regret, formally (2/3)

What is the payoff/value of an infinite play?

Discounted-Sum

We aggregate the infinite weight sequence $\chi = x_0 x_1 \ldots$ with: $\sum_{i=0}^{\infty} \lambda^i x_i$, where $0 < \lambda < 1$ is a rational given as part of the input.

How do players determine a play?

Regret, formally (2/3)

What is the payoff/value of an infinite play?

Discounted-Sum

We aggregate the infinite weight sequence $\chi = x_0 x_1 \ldots$ with: $\sum_{i=0}^{\infty} \lambda^i x_i$, where $0 < \lambda < 1$ is a rational given as part of the input.

How do players determine a play?

Strategies

A strategy for $\exists$ve is a function $\sigma : V^* V_\exists \to V$. The set of all her strategies is $\Sigma_\exists$. Similar definition for $\forall$dam, with set $\Sigma_\forall$.

Regret, formally (2/3)

What is the payoff/value of an infinite play?

Discounted-Sum

We aggregate the infinite weight sequence $\chi = x_0 x_1 \ldots$ with: $\sum_{i=0}^{\infty} \lambda^i x_i$, where $0 < \lambda < 1$ is a rational given as part of the input.

How do players determine a play?

Strategies

A strategy for $\exists$ve is a function $\sigma : V^* V_\exists \to V$. The set of all her strategies is $\Sigma_\exists$. Similar definition for $\forall$dam, with set $\Sigma_\forall$.

For strategies σ and τ for $\exists$ve and $\forall$dam, we write $\mathbf{Val}(\sigma, \tau)$ for the value of the resulting play.

UNIVERSITÉ LIBRE DE BRUXELLES

Regret, formally (3/3)

What is the regret of a strategy of $\exists$ve? of a game?

Regret, formally (3/3)

What is the regret of a strategy of ∃ve? of a game?

Regret

For a strategy σ of ∃ve,

$$\mathbf{reg}^{\sigma}(G) := \sup_{\tau \in \Sigma_{\forall}} \left(\sup_{\sigma' \in \Sigma_{\exists}} \mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau) \right)$$

Regret, formally (3/3)

What is the regret of a strategy of ∃ve? of a game?

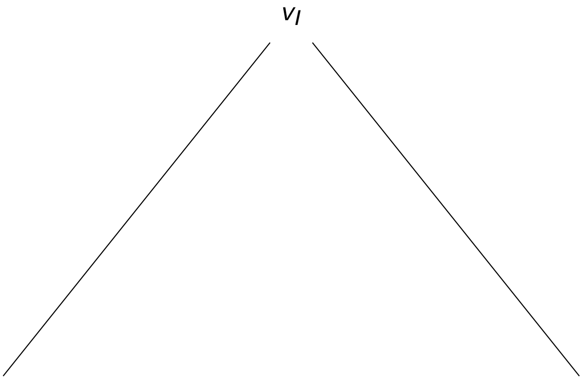
Regret

For a strategy σ of ∃ve,

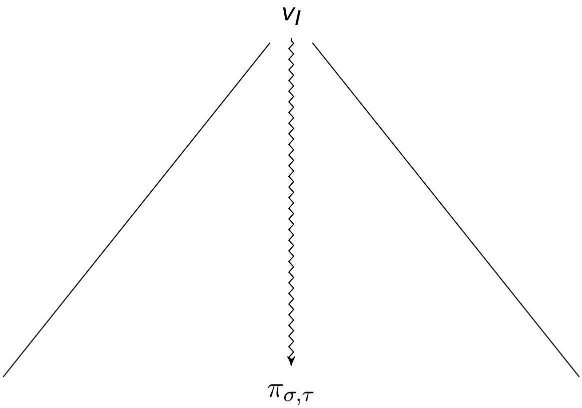
$$\mathbf{reg}^{\sigma}(G) := \sup_{\tau \in \Sigma_\forall} \left(\sup_{\sigma' \in \Sigma_\exists} \mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau) \right)$$

For a game G we have $\mathbf{Reg}(G) := \inf_{\sigma \in \Sigma_\exists} \mathbf{reg}^{\sigma}(G)$.

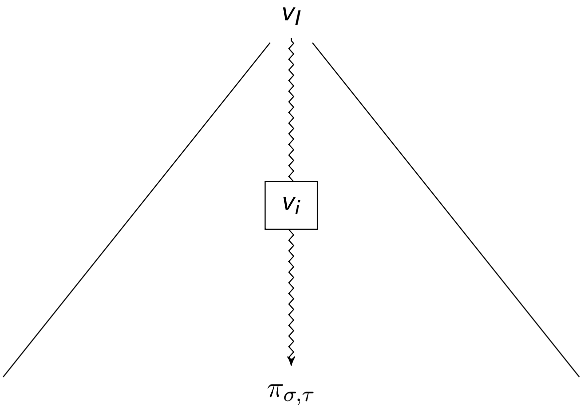
Intuitively...



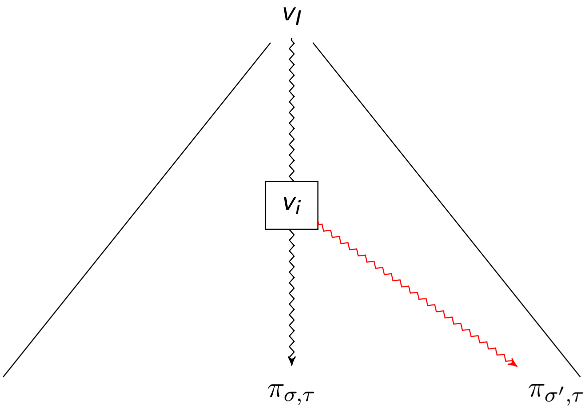
Intuitively...



Intuitively. . .



Intuitively...



Decision problems

0-regret problem

Given a game G, is it the case that $\mathbf{Reg}(G) = 0$?

Decision problems

0-regret problem

Given a game G, is it the case that $\mathbf{Reg}(G) = 0$?

If it is the case, can we construct a strategy σ for ∃ve such that $\mathbf{reg}^{\sigma}(G) = 0$?

UNIVERSITÉ LIBRE DE BRUXELLES

Decision problems

0-regret problem

Given a game G, is it the case that $\mathbf{Reg}(G) = 0$?

If it is the case, can we construct a strategy σ for ∃ve such that $\mathbf{reg}^{\sigma}(G) = 0$?

Regret threshold problem

Given a game G and a regret threshold r, is it the case that $\mathbf{Reg}(G) \leq r$?

UNIVERSITÉ LIBRE DE BRUXELLES

Decision problems

0-regret problem

Given a game G, is it the case that $\mathbf{Reg}(G) = 0$?

If it is the case, can we construct a strategy σ for ∃ve such that $\mathbf{reg}^{\sigma}(G) = 0$?

Regret threshold problem

Given a game G and a regret threshold r, is it the case that $\mathbf{Reg}(G) \leq r$?

If it is the case, can we construct a strategy σ for ∃ve such that $\mathbf{reg}^{\sigma}(G) \leq r$?

This talk in a slide

	Any strategy	Memoryless strategies	Word strategies
regret threshold	NP	PSPACE, coNP-h	PSPACE-c (ϵ-gap)
0-regret	P	PSPACE	NP-c

Table: Complexity of deciding the regret threshold and 0-regret problems for fixed λ.

Additional defs. for Discounted-Sum

The antagonistic and co-operative values of a game from vertex u:

Additional defs. for Discounted-Sum

The antagonistic and co-operative values of a game from vertex u:

Game values

$$\mathbf{aVal}^u(G) := \sup_{\sigma \in \Sigma_\exists} \inf_{\tau \in \Sigma_\forall} \mathbf{Val}(\sigma, \tau)$$

$$\mathbf{cVal}^u(G) := \sup_{\sigma \in \Sigma_\exists} \sup_{\tau \in \Sigma_\forall} \mathbf{Val}(\sigma, \tau)$$

Additional defs. for Discounted-Sum

The antagonistic and co-operative values of a game from vertex u:

Game values

$$\mathbf{aVal}^u(G) := \sup_{\sigma \in \Sigma_\exists} \inf_{\tau \in \Sigma_\forall} \mathbf{Val}(\sigma, \tau)$$

$$\mathbf{cVal}^u(G) := \sup_{\sigma \in \Sigma_\exists} \sup_{\tau \in \Sigma_\forall} \mathbf{Val}(\sigma, \tau)$$

Additionally $\mathbf{cVal}^u_{\neg v}(G)$ is the co-operative value from u when we cannot go to v as a first step.

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Additional defs. for Discounted-Sum

The antagonistic and co-operative values of a game from vertex u:

Game values

$$\mathbf{aVal}^u(G) := \sup_{\sigma \in \Sigma_\exists} \inf_{\tau \in \Sigma_\forall} \mathbf{Val}(\sigma, \tau)$$

$$\mathbf{cVal}^u(G) := \sup_{\sigma \in \Sigma_\exists} \sup_{\tau \in \Sigma_\forall} \mathbf{Val}(\sigma, \tau)$$

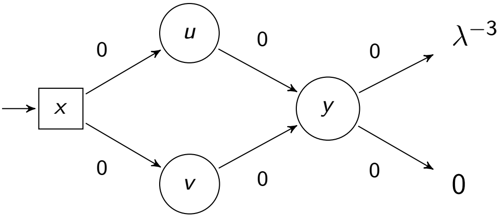
Additionally $\mathbf{cVal}^u_{\neg v}(G)$ is the co-operative value from u when we cannot go to v as a first step.

Zwick and Paterson '96

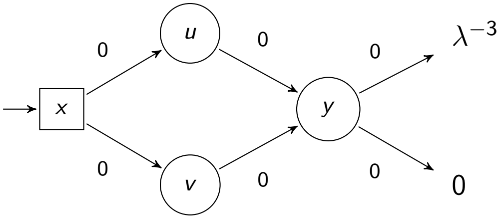
Discounted-Sum games are positionally determined, i.e. players have positional strategies which ensure **aVal**. Both **cVal** and **aVal** are computable.

Also, **cVal** and **aVal** are representable using a poly number of bits.

Deviations, an example



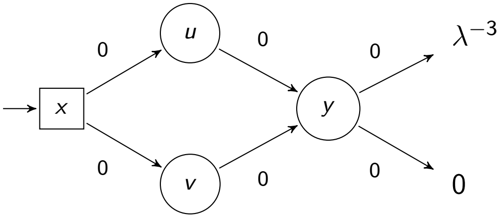
Deviations, an example



$\mathbf{cVal}^x = 1$, $\mathbf{aVal}^x = \mathbf{aVal}^u = \mathbf{aVal}^v = 0$,

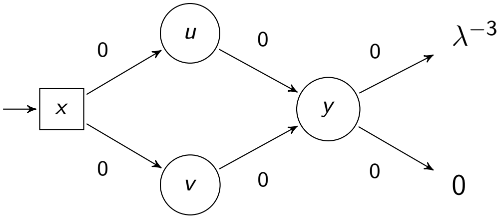
UNIVERSITÉ LIBRE DE BRUXELLES

Deviations, an example



$\mathbf{cVal}^x = 1$, $\mathbf{aVal}^x = \mathbf{aVal}^u = \mathbf{aVal}^v = 0$,
$\mathbf{cVal}^x_{\neg u} = \mathbf{cVal}^x_{\neg v} = 1$

Deviations, an example



$\mathbf{cVal}^x = 1$, $\mathbf{aVal}^x = \mathbf{aVal}^u = \mathbf{aVal}^v = 0$,
$\mathbf{cVal}^x_{\neg u} = \mathbf{cVal}^x_{\neg v} = 1$
The regret of the game is 1.

The 0-regret problem

Does the game have regret 0?

The 0-regret problem

Does the game have regret 0?

A new safety game $\hat{G}$

The game $\hat{G}$ is played on G without w. The set of unsafe edges is $\mathcal{B} := \{(u, v) \in E : u \in V_\exists \text{ and } w(u, v) + \lambda \mathbf{aVal}^v(G) < \mathbf{cVal}^u_{\neg v}(G)\}$.

Intuitively, vertices of ∃ve where every choice has a "better alternative" are bad.

The 0-regret problem

Does the game have regret 0?

A new safety game $\hat{G}$

The game $\hat{G}$ is played on G without w. The set of unsafe edges is $\mathcal{B} := \{(u, v) \in E : u \in V_\exists \text{ and } w(u, v) + \lambda \mathbf{aVal}^v(G) < \mathbf{cVal}^u_{\neg v}(G)\}$.

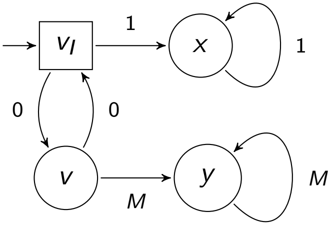
Intuitively, vertices of $\exists$ve where every choice has a "better alternative" are bad.

Lemma

G has regret 0 iff $\exists$ve has a regret-free strategy iff she has a winning strategy in the safety game $\hat{G}$.

UNIVERSITÉ LIBRE DE BRUXELLES

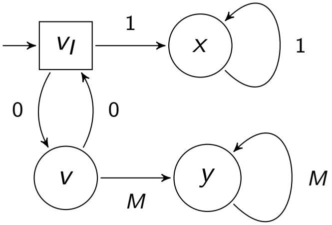
Waiting to minimize regret



For $\lambda M > 1$,

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Waiting to minimize regret



For $\lambda M > 1$, ∃ve can ensure a regret of at most $\frac{1-\lambda^{2N}}{1-\lambda}$ by N times vertex v, for $N \geq \frac{-\log M}{2\log\lambda} - \frac{1}{2}$ (so that $\lambda^{2N+1}M \leq 1$).

The regret threshold problem

Does the game have regret at most r?

Lemma

If ∃ve has no regret-free strategy, $\hat{G}$ gives us a lower bound for the regret of G: the minimal (discounted) bad edge.

The regret threshold problem

Does the game have regret at most r?

Lemma

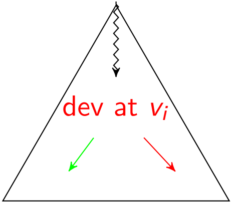
If $\exists$ve has no regret-free strategy, $\hat{G}$ gives us a lower bound for the regret of G: the minimal (discounted) bad edge.

Lemma

If the regret of σ is $0 < \rho$, then some alternative σ' necessarily deviates in at most a pseudo-polynomial number of turns (w.r.t. λ).

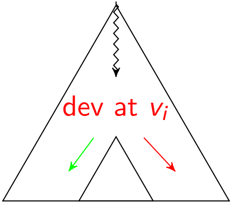
The algorithm

For some $\pi = v_0 \ldots$ consistent with σ and σ' we have $\sigma(v_0 \ldots v_i) \neq \sigma'(v_0 \ldots v_i)$



The algorithm

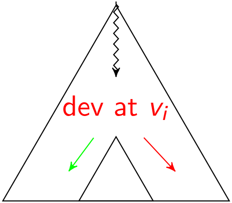
For some $\pi = v_0 \ldots$ consistent with σ and σ' we have $\sigma(v_0 \ldots v_i) \neq \sigma'(v_0 \ldots v_i)$



$$\begin{aligned} \text{regret} &\leq \lambda^i \cdot (\mathbf{cVal}^{v_i}(G) - \mathbf{aVal}^{v_i}(G)) \\ &\leq \lambda^i \cdot \frac{2W}{1-\lambda} \end{aligned}$$

The algorithm

For some $\pi = v_0 \ldots$ consistent with σ and σ' we have $\sigma(v_0 \ldots v_i) \neq \sigma'(v_0 \ldots v_i)$

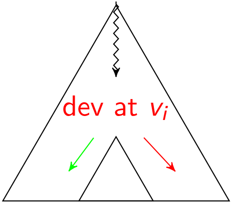


$$\begin{aligned} \text{regret} &\leq \lambda^i \cdot (\mathbf{cVal}^{v_i}(G) - \mathbf{aVal}^{v_i}(G)) \\ &\leq \lambda^i \cdot \frac{2W}{1-\lambda} \end{aligned}$$

Using the lower bound from $\hat{G}$, compute i after which any alternative σ' yields less regret. Simulate the game up to i and assume ∃ve plays for **aVal** afterwards.

The algorithm

For some $\pi = v_0 \dots$ consistent with σ and σ' we have $\sigma(v_0 \dots v_i) \neq \sigma'(v_0 \dots v_i)$



$$\begin{aligned} \text{regret} &\leq \lambda^i \cdot (\mathbf{cVal}^{v_i}(G) - \mathbf{aVal}^{v_i}(G)) \\ &\leq \lambda^i \cdot \frac{2W}{1-\lambda} \end{aligned}$$

Using the lower bound from $\hat{G}$, compute i after which any alternative σ' yields less regret. Simulate the game up to i and assume ∃ve plays for **aVal** afterwards.

Theorem

Determining if G has regret at most r is in AEXPTIME = EXPSPACE.

Simple regret-minimizing behaviors

Theorem

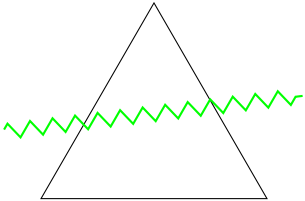
If $\exists$ve has a strategy to ensure at most regret r, then she has one which is co-operative and then antagonistic.

Simple regret-minimizing behaviors

Theorem

If ∃ve has a strategy to ensure at most regret r, then she has one which is co-operative and then antagonistic.

The unfolding of the game looks like. . .



Where, above the line she plays a positional co-operative strategy and below a positional antagonistic strategy.

UNIVERSITÉ LIBRE DE BRUXELLES

Simple regret-minimizing behaviors

Theorem

If λ is not part of the input, computing the regret of such a strategy can be done in polynomial time.

Simple regret-minimizing behaviors

Theorem

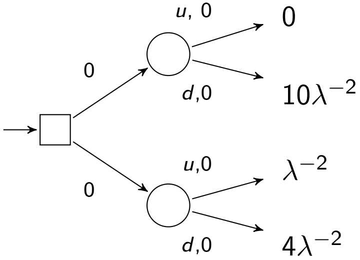
If λ is not part of the input, computing the regret of such a strategy can be done in polynomial time.

Corollary

Determining if G has regret at most r is in NP if λ is not part of the input.

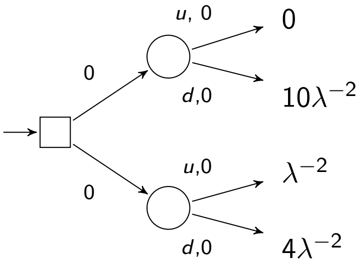
Revisitting the first example

What if the adversary is "oblivious"?



Revisitting the first example

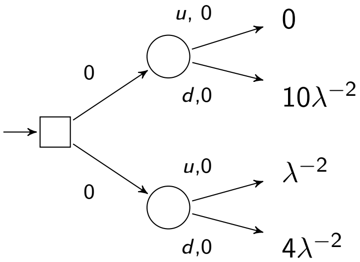
What if the adversary is "oblivious"?



We can make sure our regret is at most 1.

Revisitting the first example

What if the adversary is “oblivious”?



We can make sure our regret is at most 1.
(In DS games, models regret in portfolio investments, where all stocks are affected by the same interest rate)

Regret, against an eloquent adversary

What do we play on?

Regret, against an eloquent adversary

What do we play on?

Weighted Automata

$\mathcal{A} = (Q, q_I, A, \Delta, w)$ where (Q, q_I, A, Δ) is a finite automaton and $w : \Delta \to \mathbb{Q}$ is a weight function.

Regret, against an eloquent adversary

What do we play on?

Weighted Automata

$\mathcal{A} = (Q, q_I, A, \Delta, w)$ where (Q, q_I, A, Δ) is a finite automaton and $w : \Delta \to \mathbb{Q}$ is a weight function.

We consider infinite runs generated by $\forall$dam choosing letters and $\exists$ve resolving non-determinism

UNIVERSITÉ LIBRE DE BRUXELLES

Regret, against an eloquent adversary

What do we play on?

Weighted Automata

$\mathcal{A} = (Q, q_I, A, \Delta, w)$ where (Q, q_I, A, Δ) is a finite automaton and $w : \Delta \to \mathbb{Q}$ is a weight function.

We consider infinite runs generated by $\forall$dam choosing letters and $\exists$ve resolving non-determinism

Theorem [HPR15]

$\exists$ve has a strategy to ensure regret at most r iff the automaton is "approximate good-for-games" iff some refinement of it is r-determinizable-by-pruning.

UNIVERSITÉ LIBRE DE BRUXELLES

Algorithms

Theorem: ε-gap (promise) problem

Assuming that the regret will be $\leq r$ or $> r + \varepsilon$, determining if ∃ve has a strategy which ensures regret at most r is in EXP.[a]

[a] PSPACE if λ is not part...

Algorithms

Theorem: ε-gap (promise) problem

Assuming that the regret will be $\leq r$ or $> r + \varepsilon$, determining if ∃ve has a strategy which ensures regret at most r is in EXP.[a]

[a] PSPACE if λ is not part...

Using the approximate determinization technique of Boker & Henzinger, we can turn the regret game into an ϵ-approx Discounted-Sum game.

Algorithms

Theorem: ε-gap (promise) problem

Assuming that the regret will be $\leq r$ or $> r + \varepsilon$, determining if ∃ve has a strategy which ensures regret at most r is in EXP.[a]

[a] PSPACE if λ is not part. . .

Using the approximate determinization technique of Boker & Henzinger, we can turn the regret game into an ϵ-approx Discounted-Sum game.

Lemma

∃ve can ensure a payoff of at most $-r$ in the DS game $\hat{\mathcal{A}}$ iff she has a strategy in $\mathcal{A}$ to ensure regret at most r.

UNIVERSITÉ LIBRE DE BRUXELLES

Conclusions

Summary:

- we have given algorithms for regret threshold problems and synthesis of regret-minimizing strategies;
- in the first case, we know the problem is harder than solving DS games;
- we have considered regret-minimization against positional and eloquent adversaries (both PSPACE-hard even if λ is part. . .).

	Any strategy	Memoryless strategies	Word strategies
regret threshold	NP	PSPACE, coNP-h	PSPACE-c (ϵ-gap)
regret-free	P	PSPACE	NP-c

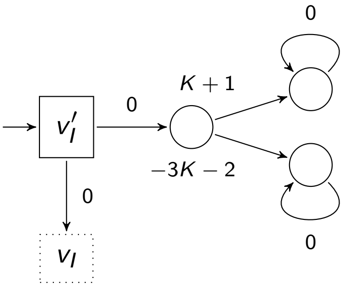
UNIVERSITÉ LIBRE DE BRUXELLES

The End

Brenguier, R., Clemente, L., Hunter, P., Pérez, G. A., Randour, M., Raskin, J.-F., Sankur, O., and Sassolas, M. (2016). Non-zero sum games for reactive synthesis. In Dediu, A., Janousek, J., Martín-Vide, C., and Truthe, B., editors, LATA, volume 9618 of LNCS, pages 3–23. Springer.

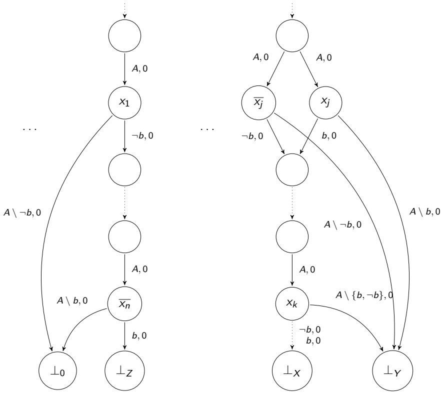
Hunter, P., Pérez, G. A., and Raskin, J.-F. (2015). Reactive synthesis without regret. Acta Informatica, pages 1–37.

DS game hardness



Gadget to reduce a game to its regret game.

PSPACE hardness



Left and right sub-arenas of the reduction from QBF. Clause i shown on the left; existential and universal gadgets for variables x_j and x_k, respectively, on the right.