

Quantitative Sabotage Games

Paul Hunter

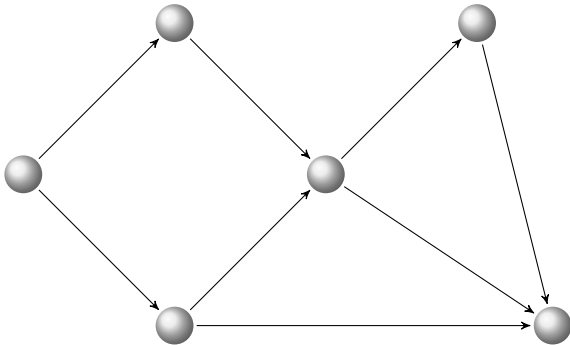
Université Libre de Bruxelles

Work of: Thomas Brihaye, Gilles Geeraerts, Axel Haddad,
Benjamin Monmege, Guillermo Pérez, Gabriel Renault

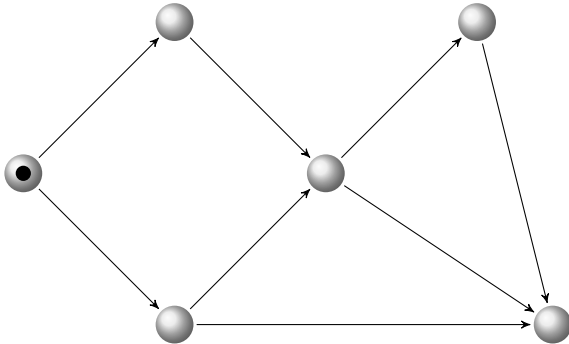
GRASTA

October 2015

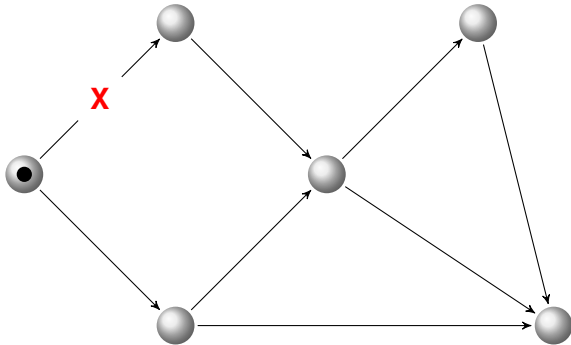
Sabotage games



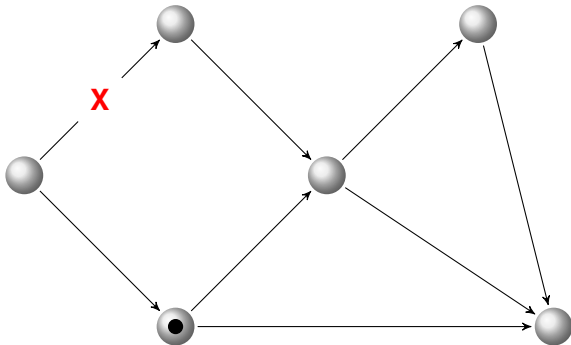
Sabotage games



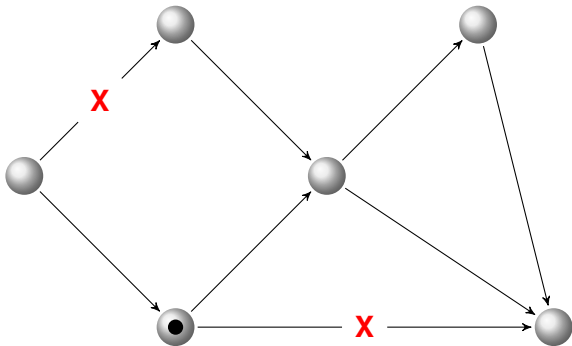
Sabotage games



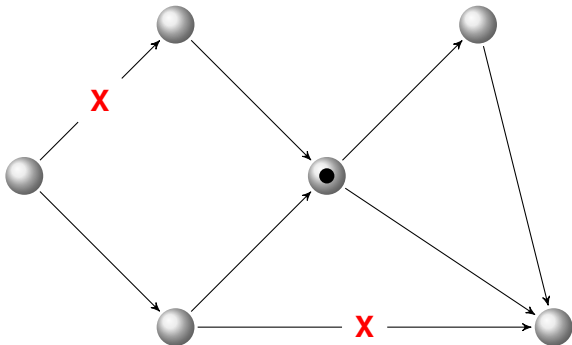
Sabotage games



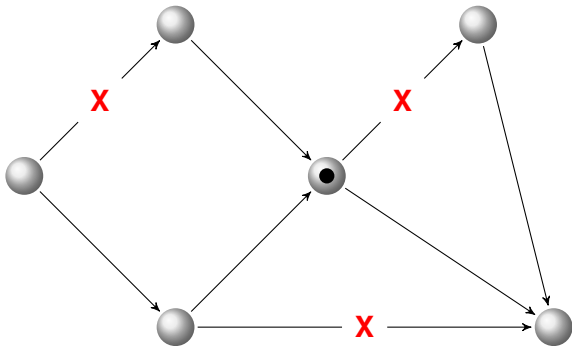
Sabotage games



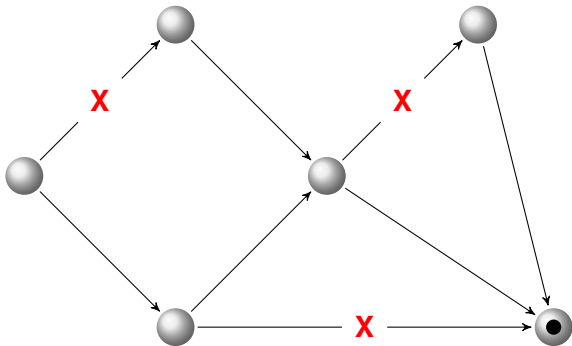
Sabotage games



Sabotage games



Sabotage games



Sabotage games

Van Benthem (2005):

- ▶ Reachability constraint
- ▶ PSPACE-complete (Löding and Rohde)

Kurzen (2011):

- ▶ Safety constraint (stay alive forever)
- ▶ PSPACE-complete

⇒ “budgeting” for saboteur

Sabotage games

Van Benthem (2005):

- ▶ Reachability constraint
- ▶ PSPACE-complete (Löding and Rohde)

Kurzen (2011):

- ▶ Safety constraint (stay alive forever)
- ▶ PSPACE-complete

⇒ “budgeting” for saboteur

Sabotage games

Van Benthem (2005):

- ▶ Reachability constraint
- ▶ PSPACE-complete (Löding and Rohde)

Kurzen (2011):

- ▶ Safety constraint (stay alive forever)
- ▶ PSPACE-complete

⇒ “budgeting” for saboteur

Quantitative Sabotage Games

In this talk (BGHMPR, FSTTCS 2015):

- ▶ Add dynamism (faults move)
- ▶ Quantitative objectives (faults penalize)

E.g. inf, sup, liminf, limsup, mean-payoff, discounted-sum, ...
(Saboteur = maximizer, runner = minimizer)

Quantitative Sabotage Games

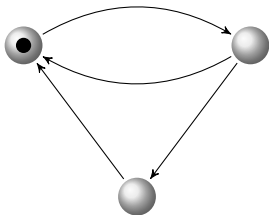
In this talk (BGHMPR, FSTTCS 2015):

- ▶ Add dynamism (faults move)
- ▶ Quantitative objectives (faults penalize)

E.g. \inf , \sup , \liminf , \limsup , mean-payoff, discounted-sum, ...
(Saboteur = maximizer, runner = minimizer)

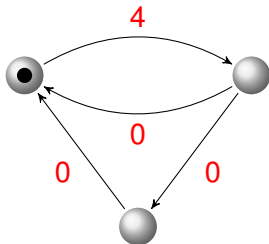
Quantitative Sabotage Games

$B = 4$, mean-payoff:



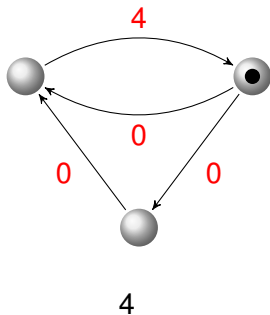
Quantitative Sabotage Games

$B = 4$, mean-payoff:



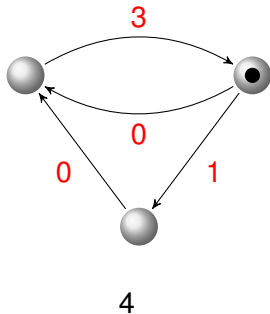
Quantitative Sabotage Games

$B = 4$, mean-payoff:



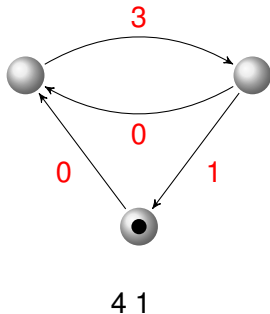
Quantitative Sabotage Games

$B = 4$, mean-payoff:



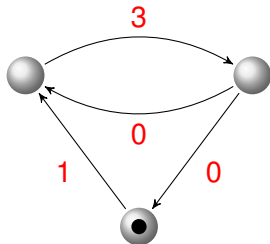
Quantitative Sabotage Games

$B = 4$, mean-payoff:



Quantitative Sabotage Games

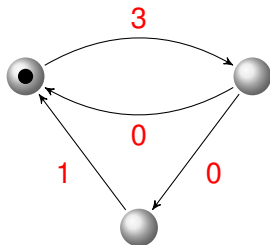
$B = 4$, mean-payoff:



4 1

Quantitative Sabotage Games

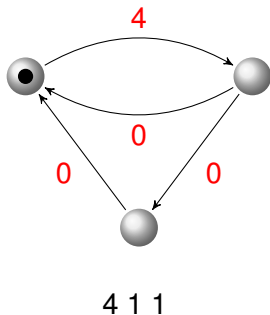
$B = 4$, mean-payoff:



4 1 1

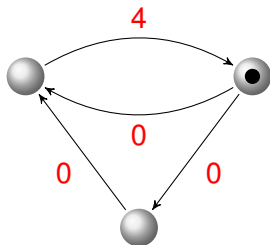
Quantitative Sabotage Games

$B = 4$, mean-payoff:



Quantitative Sabotage Games

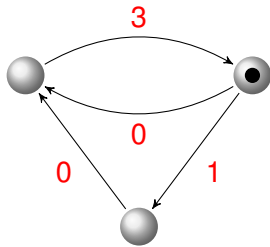
$B = 4$, mean-payoff:



4 1 1 4

Quantitative Sabotage Games

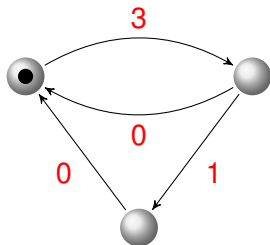
$B = 4$, mean-payoff:



4 1 1 4

Quantitative Sabotage Games

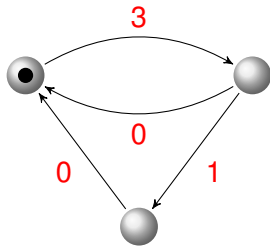
$B = 4$, mean-payoff:



4 1 1 4 0...

Quantitative Sabotage Games

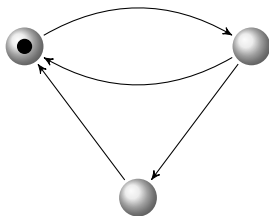
$B = 4$, mean-payoff: 2



4 1 1 4 0...

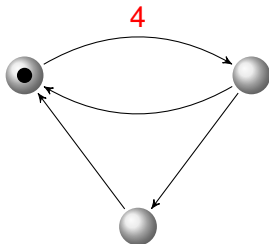
Quantitative Sabotage Games

$B = 4$, sup:



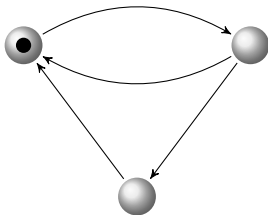
Quantitative Sabotage Games

$B = 4$, sup: 4



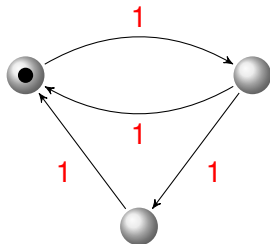
Quantitative Sabotage Games

$B = 4$, inf:



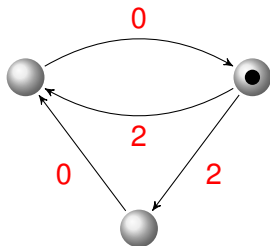
Quantitative Sabotage Games

$B = 4$, inf:



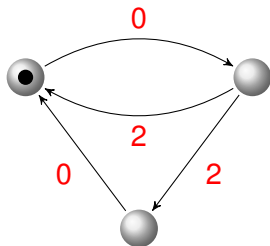
Quantitative Sabotage Games

$B = 4$, inf:



Quantitative Sabotage Games

$B = 4$, inf: 1



Complexity of QSGs

Looking at **threshold decision problem**: Is the payoff at most T ?

(e.g. sup threshold with $T = 0$ corresponds to cops and robber)

Theorem (Brihaye et al)

The threshold problem for sup, limsup, mean-payoff, and discounted-sum QSGs is EXPTIME-complete.

Complexity of QSGs

Looking at **threshold decision problem**: Is the payoff at most T ?

(e.g. sup threshold with $T = 0$ corresponds to cops and robber)

Theorem (Brihaye et al)

The threshold problem for sup, limsup, mean-payoff, and discounted-sum QSGs is EXPTIME-complete.

EXPTIME-hardness

Reduction from ALTERNATING BOOLEAN FORMULA (ABF) to
EXTENDED SAFETY GAME

EXPTIME-hardness

Alternating Boolean Formula:

- ▶ Given: formula φ (in CNF), truth assignment α , and a partition of the variables of φ (X, Y)
- ▶ `Prover` and `Disprover` alternately change α by changing the truth value of some variable in their partition
- ▶ `Prover` wins if φ is ever true under α .

Shown to be EXPTIME-complete by Stockmeyer and Chandra (1979).

Extended Safety Game:

- ▶ QSG with sup payoff and threshold 0
- ▶ “Safe” edges which cannot be occupied by saboteur
- ▶ “Final” vertices which terminate the game if reached by runner, winning for runner iff not occupied by saboteur.

EXPTIME-hardness

Alternating Boolean Formula:

- ▶ Given: formula φ (in CNF), truth assignment α , and a partition of the variables of φ (X, Y)
- ▶ `Prover` and `Disprover` alternately change α by changing the truth value of some variable in their partition
- ▶ `Prover` wins if φ is ever true under α .

Shown to be EXPTIME-complete by Stockmeyer and Chandra (1979).

Extended Safety Game:

- ▶ QSG with sup payoff and threshold 0
- ▶ “Safe” edges which cannot be occupied by saboteur
- ▶ “Final” vertices which terminate the game if reached by runner, winning for runner iff not occupied by saboteur.

EXPTIME-hardness: Overview

- ▶ **Saboteur** = `Prover`
- ▶ Two final vertices for each literal (i.e. four per literal pair). Occupied vertices indicate the current truth assignment.
 - ▶ Gadget forces *at least* two occupied per literal pair
 - ▶ Budget forces *at most* two occupied per literal pair
- ▶ Runner sets his variables by threatening unoccupied final vertices.
- ▶ Non-threatening moves let saboteur set his variables. Runner ensures correct variables are changed.
- ▶ Saboteur can move to a terminating path which ends in an occupied final vertex iff all clauses are satisfied.

EXPTIME-hardness: Overview

- ▶ Saboteur = Prover
- ▶ Two final vertices for each literal (i.e. four per literal pair). Occupied vertices indicate the current truth assignment.
 - ▶ Gadget forces *at least* two occupied per literal pair
 - ▶ Budget forces *at most* two occupied per literal pair
- ▶ Runner sets his variables by threatening unoccupied final vertices.
- ▶ Non-threatening moves let saboteur set his variables. Runner ensures correct variables are changed.
- ▶ Saboteur can move to a terminating path which ends in an occupied final vertex iff all clauses are satisfied.

EXPTIME-hardness: Overview

- ▶ Saboteur = `Prover`
- ▶ Two final vertices for each literal (i.e. four per literal pair). Occupied vertices indicate the current truth assignment.
 - ▶ Gadget forces *at least* two occupied per literal pair
 - ▶ Budget forces *at most* two occupied per literal pair
- ▶ Runner sets his variables by threatening unoccupied final vertices.
- ▶ Non-threatening moves let saboteur set his variables. Runner ensures correct variables are changed.
- ▶ Saboteur can move to a terminating path which ends in an occupied final vertex iff all clauses are satisfied.

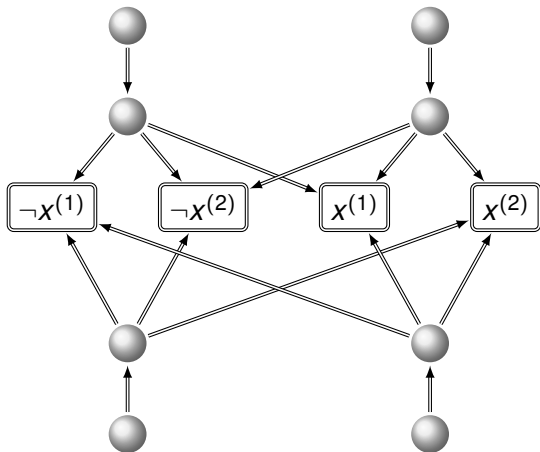
EXPTIME-hardness: Overview

- ▶ Saboteur = `Prover`
- ▶ Two final vertices for each literal (i.e. four per literal pair). Occupied vertices indicate the current truth assignment.
 - ▶ Gadget forces *at least* two occupied per literal pair
 - ▶ Budget forces *at most* two occupied per literal pair
- ▶ Runner sets his variables by threatening unoccupied final vertices.
- ▶ Non-threatening moves let saboteur set his variables. Runner ensures correct variables are changed.
- ▶ Saboteur can move to a terminating path which ends in an occupied final vertex iff all clauses are satisfied.

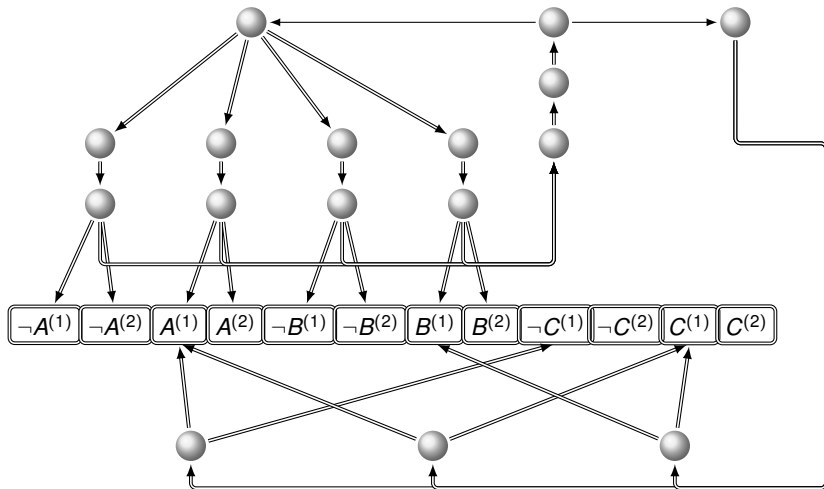
EXPTIME-hardness: Overview

- ▶ Saboteur = `Prover`
- ▶ Two final vertices for each literal (i.e. four per literal pair). Occupied vertices indicate the current truth assignment.
 - ▶ Gadget forces *at least* two occupied per literal pair
 - ▶ Budget forces *at most* two occupied per literal pair
- ▶ Runner sets his variables by threatening unoccupied final vertices.
- ▶ Non-threatening moves let saboteur set his variables. Runner ensures correct variables are changed.
- ▶ Saboteur can move to a terminating path which ends in an occupied final vertex iff all clauses are satisfied.

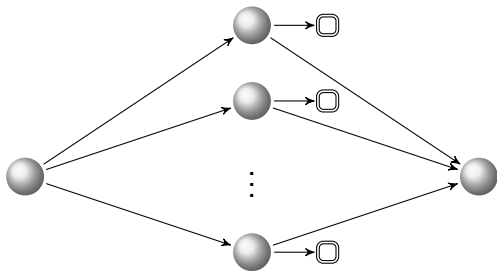
EXPTIME-hardness: Literal gadget



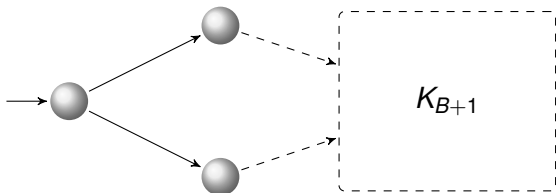
EXPTIME-hardness: General construction



EXPTIME-hardness: Safe edge gadget



EXPTIME-hardness: Final vertex gadget



Conclusions and further work

- ▶ Added quantitative goals to cops and robber
- ▶ EXPTIME-completeness for all variants (on directed graphs)

- ▶ Connection with standard cops and robber?!?
- ▶ Partial information games
- ▶ Randomized saboteur