Patrolling Games

(Theoretical approach)

Tomáš Brázdil

Masaryk University, Brno

Joint work with M. Abbaffy, M. Abbaffy, B. Bošanský, A. Kučera, V. Řehák, T. Lamser

Security Games

The goal: Protect targets with limited security resources.

Security Games

Defender protects targets $t_1, \ldots, t_n$ by assigning security resources $r_1, \ldots, r_m$.

Attacker monitors behaviour of Defender, exploits patterns.

Security Games

Defender protects targets $t_1, \ldots, t_n$ by assigning security resources $r_1, \ldots, r_m$.

Attacker monitors behaviour of Defender, exploits patterns.

Stackelberg Security Games:
Attacker knows the strategy of Defender.

Game theory provides *randomized* assignment of resources that helps to "confuse" attackers.

Security Games

Very rich area of current applied research.

Successful real-world applications:

- ARMOR [Pita et al, AAAI, 2008]
  Los Angeles International Airport, placement of checkpoints on inbound roads and allocation of canine patrols.


- IRIS [Tsai et al, AAMAS, 2009]
  Randomized deployment of the US Federal Air Marshals.

- PROTECT [Shieh et al, AAMAS, 2012]
  Randomized scheduling of Coast Guard patrols.

- . . .

Patrolling Games

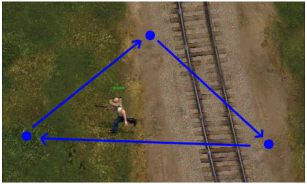
A special case of security games.

Patrolling Games

A special case of security games.

Defender moves sequentially among targets.

His moves are constrained by a graph, or by "geometry" of the situation.

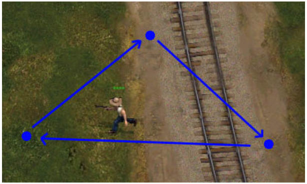


Patrolling Games

A special case of security games.

Defender moves sequentially among targets.

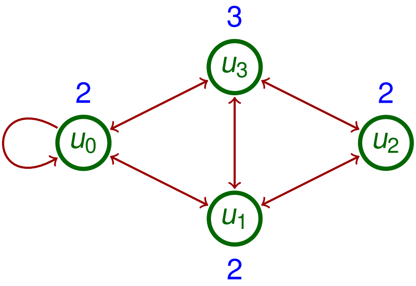
His moves are constrained by a graph, or by "geometry" of the situation.



Attacker observes the current location of Defender, and knows his future *randomized* moves.

An attack on a target takes some time during which Defender may capture Attacker.

Patrolling Games

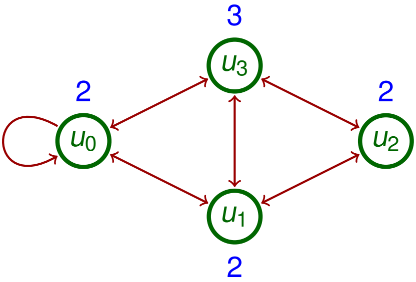


A *patrolling game* consists of

- a finite set U of *nodes*,
- an initial node u_0,
- a set E of *arcs*,
- a function $d : U \to \mathbb{N}$ assigning to every node an *attack length*.

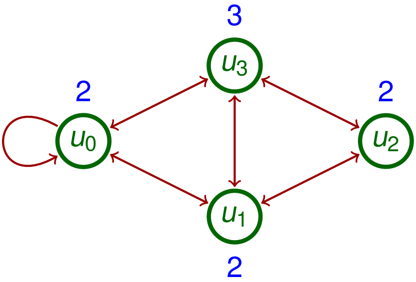
Two players: *Defender* and *Attacker*

Patrolling Games – Defender



A play starts in the initial node u_0.

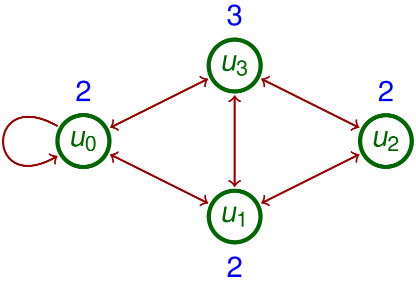
Patrolling Games – Defender



A play starts in the initial node u_0.

In every step, Defender moves to a next node chosen *randomly* according to an arbitrary probability distribution.

Patrolling Games – Defender

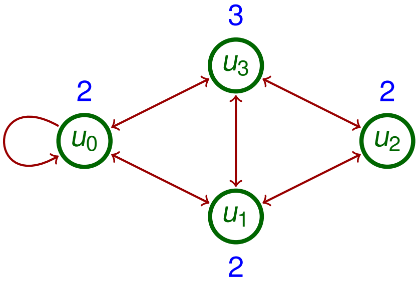


A play starts in the initial node u_0.

In every step, Defender moves to a next node chosen *randomly* according to an arbitrary probability distribution.

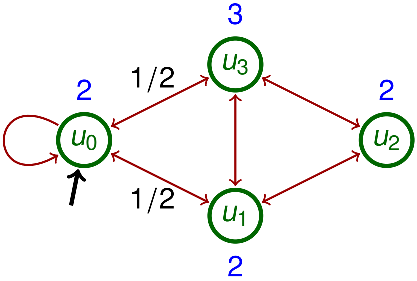
Formally, Defender's strategy σ is a function which to every history $h \in U^+$ assigns a probability distribution on U.

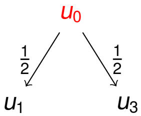
Patrolling Games – Defender



Example of a Defender's strategy σ :

Patrolling Games – Defender

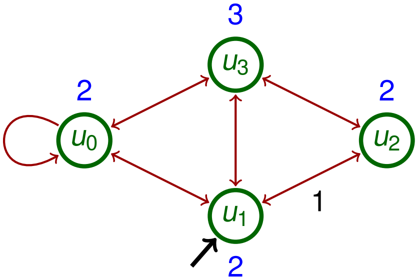


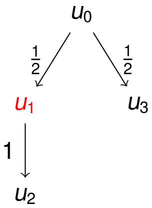


Example of a Defender's strategy σ :

- In u_0 go to u_1 with probability $1/2$, and to u_3 with probability $1/2$.

Patrolling Games – Defender

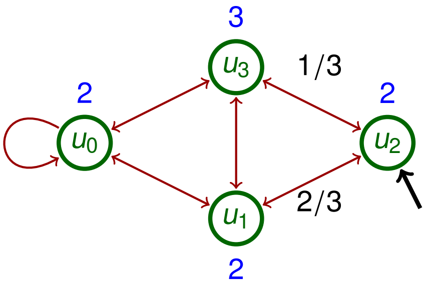


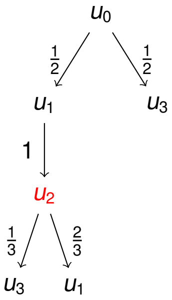


Example of a Defender's strategy σ :

- In u_0 go to u_1 with probability $1/2$, and to u_3 with probability $1/2$.
- in u_1 go to u_2 if u_0 has been visited odd number of times, else go to u_3.

Patrolling Games – Defender

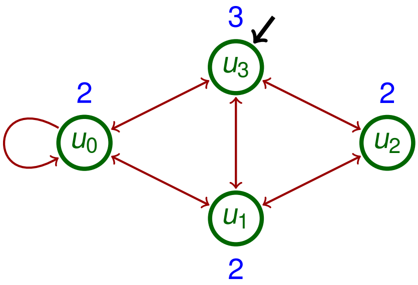


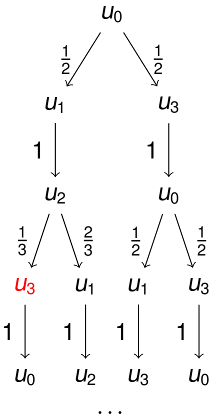


Example of a Defender's strategy σ :

- In u_0 go to u_1 with probability $1/2$, and to u_3 with probability $1/2$.
- in u_1 go to u_2 if u_0 has been visited odd number of times, else go to u_3.
- in u_2 go to u_3 with prob. $1/3$, and to u_1 with prob. $2/3$.

Patrolling Games – Defender

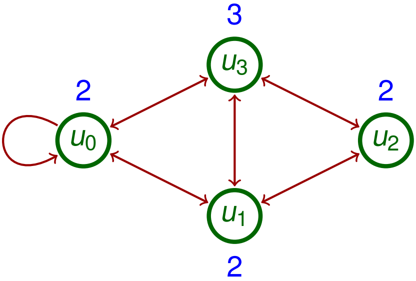




Example of a Defender's strategy σ :

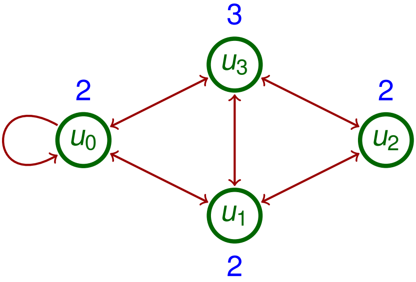
- In u_0 go to u_1 with probability $1/2$, and to u_3 with probability $1/2$.
- in u_1 go to u_2 if u_0 has been visited odd number of times, else go to u_3.
- in u_2 go to u_3 with prob. $1/3$, and to u_1 with prob. $2/3$.
- in u_3 go to u_0 with probability 1.

Patrolling Games - Attacker



In every step, Attacker may randomly decide whether to attack one of the nodes, or stay idle.

Patrolling Games - Attacker

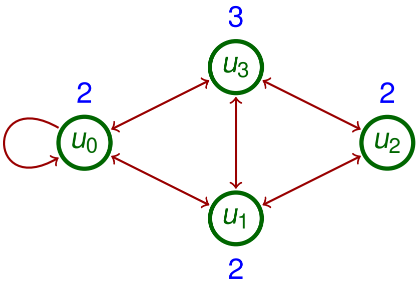


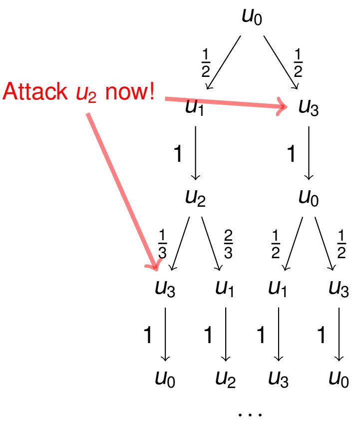
In every step, Attacker may randomly decide whether to attack one of the nodes, or stay idle.

Formally, a strategy π of Attacker is a function which to every history $h \in U^*$ assigns a distribution on $U \cup \{\mathcal{I}\}$.

Here $\mathcal{I}$ means *stay idle*.

Patrolling Games – Attacker





Attacker: Attack u_2 whenever defender is in u_3 and there was an odd number of visits to u_0 in the past, otherwise stay idle.

The Value

Fix strategies σ and π of Defender and Attacker, resp.

The Value

Fix strategies σ and π of Defender and Attacker, resp.

Attack on a node u is *successful* if Defender *does not* visit u in the next $d(u)$ steps; otherwise, the defence is successful.

Note that even if Defender is in u when the attack starts, he still has to visit u in the next $d(u)$ steps.

The Value

Fix strategies σ and π of Defender and Attacker, resp.

Attack on a node u is *successful* if Defender *does not* visit u in the next $d(u)$ steps; otherwise, the defence is successful.

Note that even if Defender is in u when the attack starts, he still has to visit u in the next $d(u)$ steps.

Attacker wins if his attack is successful.

The Value

Fix strategies σ and π of Defender and Attacker, resp.

Attack on a node u is *successful* if Defender *does not* visit u in the next $d(u)$ steps; otherwise, the defence is successful.

Note that even if Defender is in u when the attack starts, he still has to visit u in the next $d(u)$ steps.

Attacker wins if his attack is successful.

Defender wins if either there is no attack at all, or the attack is not successful.

The Value

Fix strategies σ and π of Defender and Attacker, resp.

Attack on a node u is *successful* if Defender *does not* visit u in the next $d(u)$ steps; otherwise, the defence is successful.

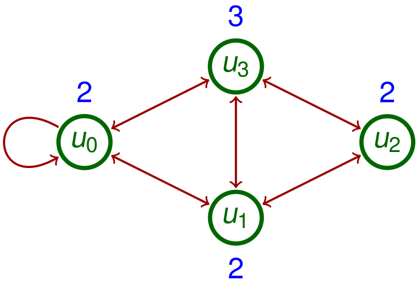
Note that even if Defender is in u when the attack starts, he still has to visit u in the next $d(u)$ steps.

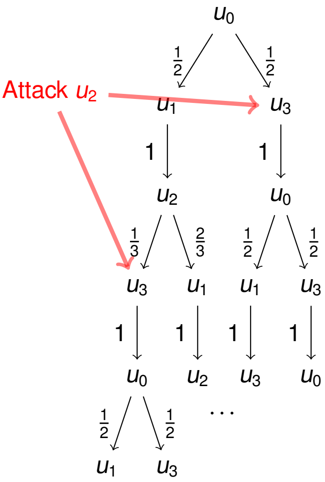
Attacker wins if his attack is successful.

Defender wins if either there is no attack at all, or the attack is not successful.

Denote by $P[\sigma, \pi]$ the probability that Defender wins.

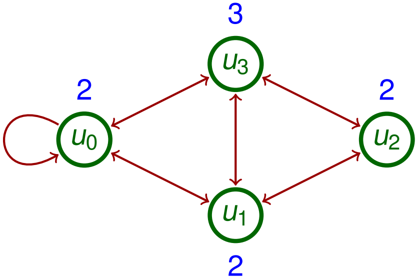
The Value

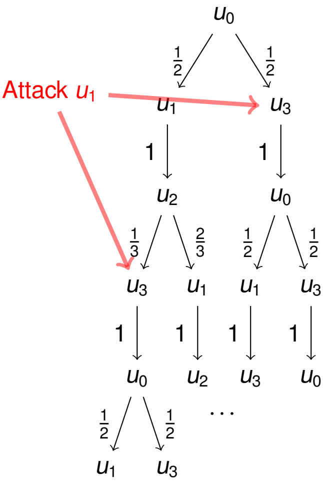




Here Attacker wins with probability one, i.e. $P[\sigma, \pi] = 0$.

The Value





Here Attacker wins with probability $\frac{1}{2}$, i.e. $P[\sigma, \pi] = \frac{1}{2}$.

Patrolling Games – Determinacy

Theorem 1
The patrolling games are determined, i.e.

$$\sup_{\sigma} \inf_{\pi} P[\sigma, \pi] = \inf_{\pi} \sup_{\sigma} P[\sigma, \pi]$$

Patrolling Games – Determinacy

Theorem 1
The patrolling games are determined, i.e.

$$\sup_{\sigma} \inf_{\pi} P[\sigma, \pi] = \inf_{\pi} \sup_{\sigma} P[\sigma, \pi]$$

Patrolling games are turn-based zero-sum games where

Patrolling Games – Determinacy

Theorem 1
The patrolling games are determined, i.e.

$$\sup_{\sigma} \inf_{\pi} P[\sigma, \pi] = \inf_{\pi} \sup_{\sigma} P[\sigma, \pi]$$

Patrolling games are turn-based zero-sum games where

- Attacker has perfect information,

Patrolling Games – Determinacy

Theorem 1
The patrolling games are determined, i.e.

$$\sup_{\sigma} \inf_{\pi} P[\sigma, \pi] = \inf_{\pi} \sup_{\sigma} P[\sigma, \pi]$$

Patrolling games are turn-based zero-sum games where

- Attacker has perfect information,
- Defender has imperfect information.

Patrolling Games – Determinacy

Theorem 1

The patrolling games are determined, i.e.

$$\sup_{\sigma} \inf_{\pi} P[\sigma, \pi] = \inf_{\pi} \sup_{\sigma} P[\sigma, \pi]$$

Patrolling games are turn-based zero-sum games where

- Attacker has perfect information,
- Defender has imperfect information.

  However, Defender is perfectly informed about moves made more than $\max_{u \in U} d(u)$ steps in the past.

Patrolling Games – Determinacy

Theorem 1
The patrolling games are determined, i.e.

$$\sup_{\sigma} \inf_{\pi} P[\sigma, \pi] = \inf_{\pi} \sup_{\sigma} P[\sigma, \pi]$$

Patrolling games are turn-based zero-sum games where

- Attacker has perfect information,
- Defender has imperfect information.

  However, Defender is perfectly informed about moves made more than $\max_{u \in U} d(u)$ steps in the past.

Patrolling games are a special case of infinite games with eventual perfect monitoring, that have been shown to be determined in [Shmaya, Proc. of AMS, 2011].

Patrolling Games – Determinacy

Theorem 1
The patrolling games are determined, i.e.

$$\sup_{\sigma} \inf_{\pi} P[\sigma,\pi] = \inf_{\pi} \sup_{\sigma} P[\sigma,\pi]$$

Patrolling games are turn-based zero-sum games where

- Attacker has perfect information,
- Defender has imperfect information.

  However, Defender is perfectly informed about moves made more than $\max_{u \in U} d(u)$ steps in the past.

Patrolling games are a special case of infinite games with eventual perfect monitoring, that have been shown to be determined in [Shmaya, Proc. of AMS, 2011].

Questions: Does defender have an optimal strategy? If yes, could it be effectively computed?

Solving Patrolling Games

Most existing solutions concentrate either

- on stationary strategies (or bounded memory strategies) and reduce either to non-linear programming, or to mixed integer linear programming,
  see e.g. [Basilico, Gatti, Amigoni, AI 2012], [Vorobeychik, An, Tambe, Singh, AAAI 2014]

Solving Patrolling Games

Most existing solutions concentrate either

- on stationary strategies (or bounded memory strategies) and reduce either to non-linear programming, or to mixed integer linear programming,
  see e.g. [Basilico, Gatti, Amigoni, AI 2012], [Vorobeychik, An, Tambe, Singh, AAAI 2014]
- or a finite horizon formulation and solve the problem of randomly choosing a patrolling route of a fixed length.
  see e.g. "Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned" by M. Tambe

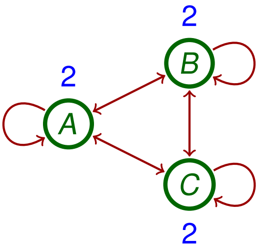
Solving Patrolling Games

Most existing solutions concentrate either

- on stationary strategies (or bounded memory strategies) and reduce either to non-linear programming, or to mixed integer linear programming,
  see e.g. [Basilico, Gatti, Amigoni, AI 2012], [Vorobeychik, An, Tambe, Singh, AAAI 2014]
- or a finite horizon formulation and solve the problem of randomly choosing a patrolling route of a fixed length.
  see e.g. "Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned" by M. Tambe

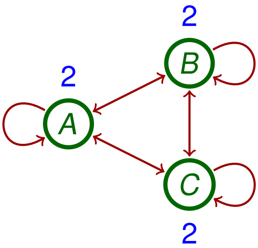
[Bošanský, Vaňek, Pěchouček, AAAI 2012] shows (experimentally) that enlarging defender's memory can be beneficial.

Example: Triangle



What is an optimal strategy σ of Defender?

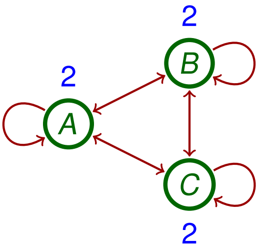
Example: Triangle



What is an optimal strategy σ of Defender?

First try: Always choose each arc with probability $\frac{1}{3}$.

Example: Triangle



What is an optimal strategy σ of Defender?

First try: Always choose each arc with probability $\frac{1}{3}$.

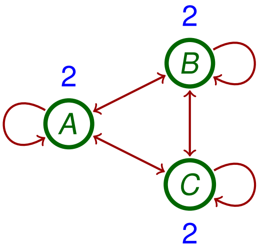
Attacker's strategy π: Attack A at the beginning.

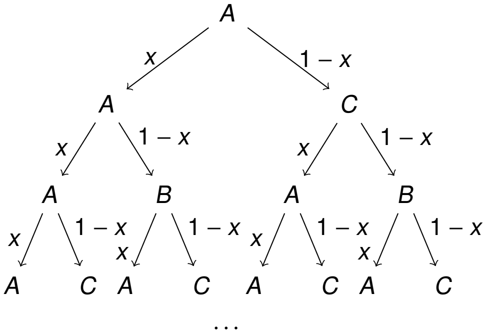
$$P[\sigma, \pi] = \frac{1}{3} + \frac{2}{3}\frac{1}{3} = \frac{5}{9}$$

All attacker's strategies that attack almost surely give the same result.

Is this optimal?

The Triangle (Cont.)



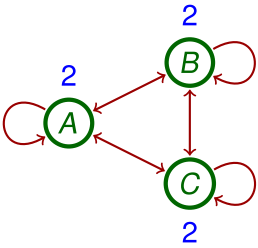


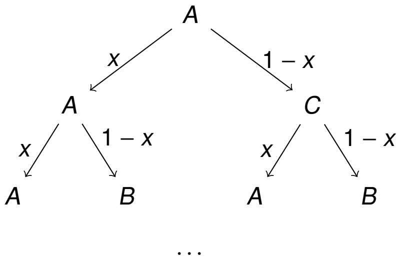
Defender's strategy σ :

In the n-th step go to

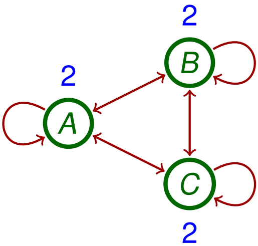
- A with probability x
- if n is even, go to B with probability $1 - x$
- if n is odd, go to C with probability $1 - x$

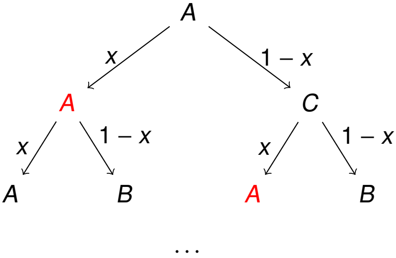
The Triangle (Cont.)





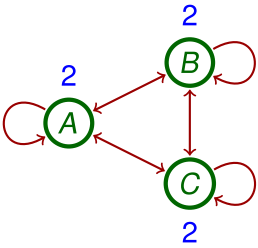
The Triangle (Cont.)

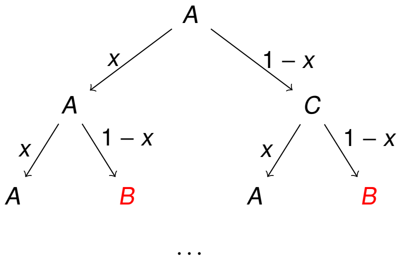




- If A is attacked, the probability of successful defence is $x + (1 - x)x = -x^2 + 2x$

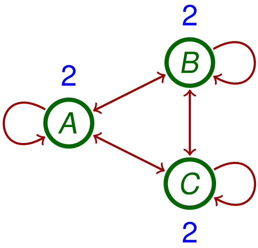
The Triangle (Cont.)

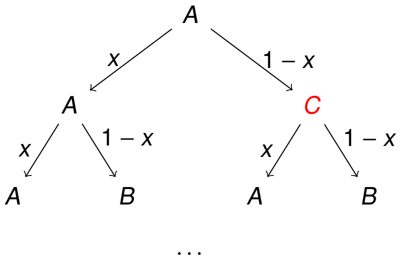




- If A is attacked, the probability of successful defence is $x + (1 - x)x = -x^2 + 2x$
- If B is attacked, the prob. of successful defence is $(1 - x)$

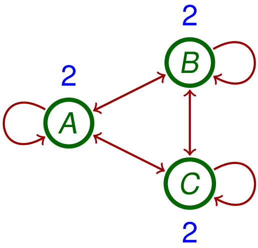
The Triangle (Cont.)

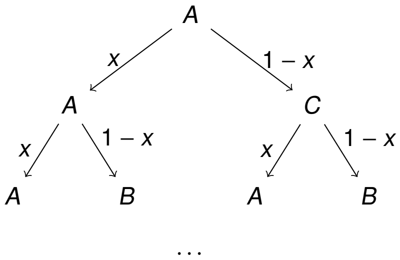




- If A is attacked, the probability of successful defence is $x + (1 - x)x = -x^2 + 2x$
- If B is attacked, the prob. of successful defence is $(1 - x)$
- If C is attacked, the prob. of successful defence is $(1 - x)$

The Triangle (Cont.)





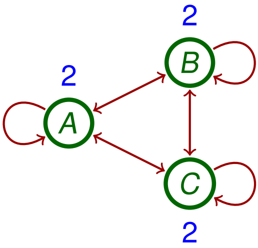
- If A is attacked, the probability of successful defence is $x + (1 - x)x = -x^2 + 2x$
- If B is attacked, the prob. of successful defence is $(1 - x)$
- If C is attacked, the prob. of successful defence is $(1 - x)$

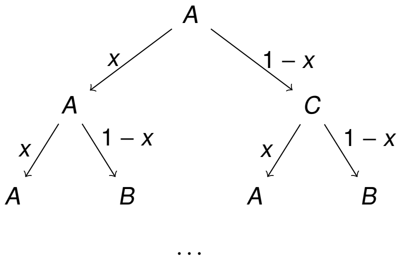
To maximize defender's probability of successful defence demand

$$-x^2 + 2x = 1 - x \qquad \text{that is} \qquad x^2 - 3x + 1 = 0$$

then $x = \frac{3-\sqrt{5}}{2}$ and the probability of successful defence is $\frac{\sqrt{5}-1}{2} \approx 0.618 > 0.56 \approx 5/9$.

The Triangle (Cont.)





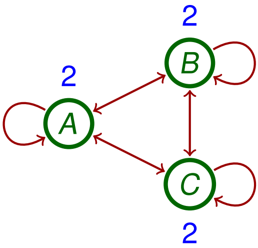
Note that defender's strategy uses "unbounded" memory.

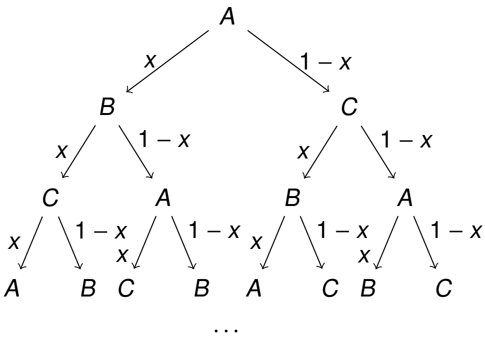
Completely ignores the actual positions in the graph.

We call such strategies *blind*.

Is it possible to construct a defender's strategy which remembers only past two steps?

The Triangle (Cont.)

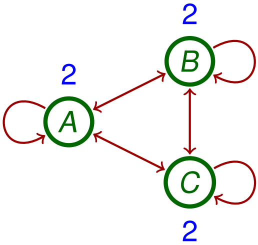


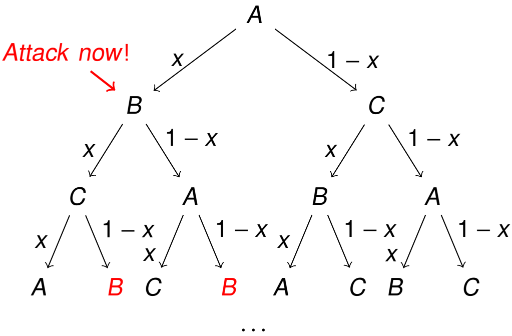


Defender's strategy σ :

- Do not go to the current node (i.e. do not take the self-loop).
- go to the previously visited node with prob. $1 - x$.
- go to the remaining node with prob. x.

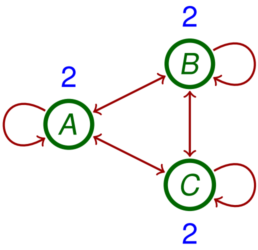
The Triangle (Cont.)

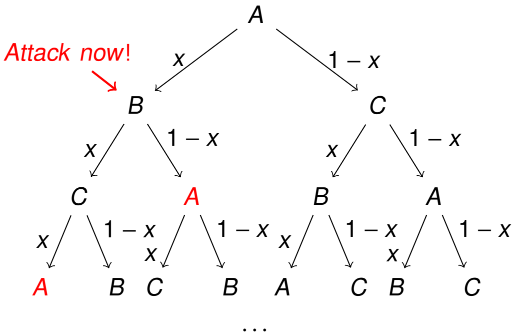




- If the current node is attacked, the prob. of defence is $1 - x$.

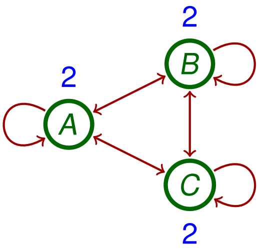
The Triangle (Cont.)

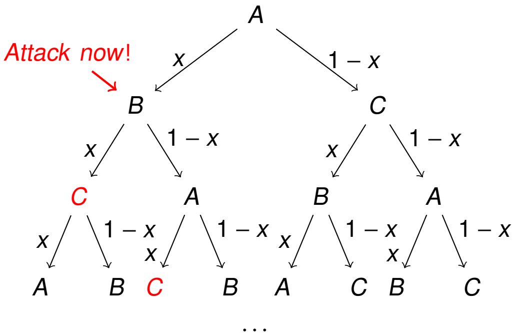




- If the current node is attacked, the prob. of defence is $1 - x$.
- If the previously visited node is attacked, the prob. of defence is $x^2 + (1 - x)$.

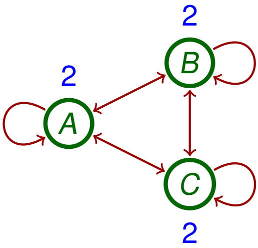
The Triangle (Cont.)

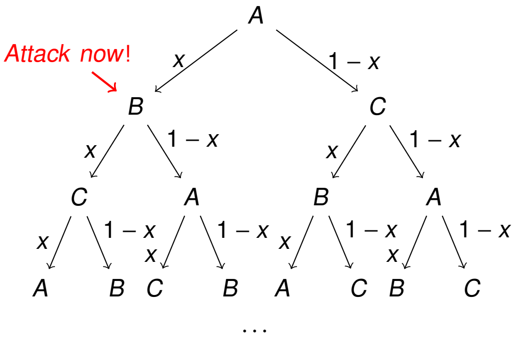




- If the current node is attacked, the prob. of defence is $1 - x$.
- If the previously visited node is attacked, the prob. of defence is $x^2 + (1 - x)$.
- If the remaining node is attacked, the prob. of defence is $x + (1 - x)x$.

The Triangle (Cont.)





- If the current node is attacked, the prob. of defence is $1 - x$.
- If the previously visited node is attacked, the prob. of defence is $x^2 + (1 - x)$.
- If the remaining node is attacked, the prob. of defence is $x + (1 - x)x$.

Then $1 - x = x + (1 - x)x$ gives $x = \frac{3-\sqrt{5}}{2}$ and $1 - x = \frac{\sqrt{5}-1}{2}$ as before.

The Triangle (Cont.)

We have two Defender's strategies:

The Triangle (Cont.)

We have two Defender's strategies:

- One uses the number of steps (modulo 2) but completely ignores actual positions in the graph.

The Triangle (Cont.)

We have two Defender's strategies:

- One uses the number of steps (modulo 2) but completely ignores actual positions in the graph.
- The other takes into account history of depth 2.

The Triangle (Cont.)

We have two Defender's strategies:

- One uses the number of steps (modulo 2) but completely ignores actual positions in the graph.
- The other takes into account history of depth 2.

Both are optimal, the value really is $\frac{\sqrt{5}-1}{2}$!

The Triangle (Cont.)

We have two Defender's strategies:

- One uses the number of steps (modulo 2) but completely ignores actual positions in the graph.
- The other takes into account history of depth 2.

Both are optimal, the value really is $\frac{\sqrt{5}-1}{2}$!

Which one may generalize to other patrolling games?

The Triangle (Cont.)

We have two Defender's strategies:

- One uses the number of steps (modulo 2) but completely ignores actual positions in the graph.
- The other takes into account history of depth 2.

Both are optimal, the value really is $\frac{\sqrt{5}-1}{2}$!

Which one may generalize to other patrolling games?

Conjecture: For every patrolling game there is an optimal Defender's strategy whose choices are determined only by the last $\max_{u \in U} d(u)$ visited nodes.

The Triangle (Cont.)

We have two Defender's strategies:

- One uses the number of steps (modulo 2) but completely ignores actual positions in the graph.
- The other takes into account history of depth 2.

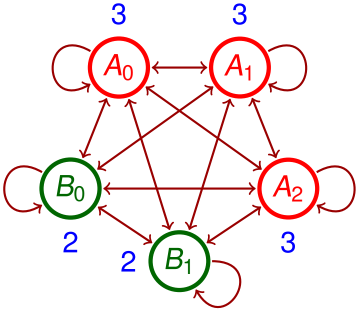
Both are optimal, the value really is $\frac{\sqrt{5}-1}{2}$!

Which one may generalize to other patrolling games?

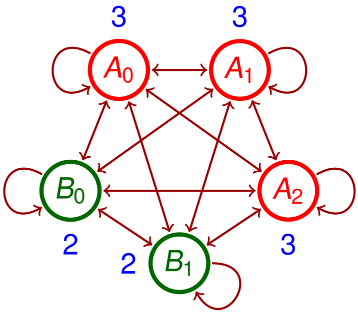
Conjecture: For every patrolling game there is an optimal Defender's strategy whose choices are determined only by the last $\max_{u \in U} d(u)$ visited nodes.

Is false!

The Pentagon



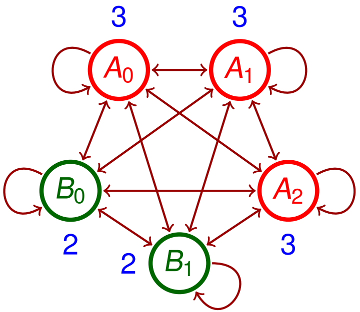
The Pentagon



Defender's strategy:
In the *n*-th step, choose uniformly between $A_{(n \mod 3)}$ and $B_{(n \mod 2)}$.

Intuition: Walk around A_i's, or B_i's, switch between these two walks with probability $1/2$ in every step.

The Pentagon

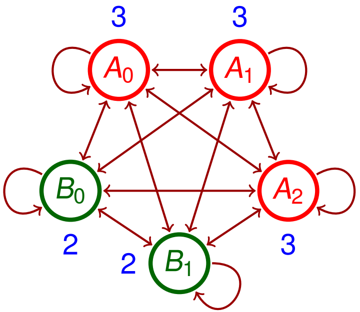


Defender's strategy:
In the n-th step, choose uniformly between $A_{(n \mod 3)}$ and $B_{(n \mod 2)}$.

Intuition: Walk around A_i's, or B_i's, switch between these two walks with probability $1/2$ in every step.

Assume that Attacker attacks a node at an arbitrary time.

The Pentagon



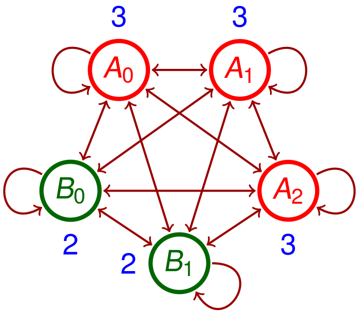
Defender's strategy:
In the *n*-th step, choose uniformly between $A_{(n \mod 3)}$ and $B_{(n \mod 2)}$.

Intuition: Walk around A_i's, or B_i's, switch between these two walks with probability $1/2$ in every step.

Assume that Attacker attacks a node at an arbitrary time.

The probability of successful defence is $\frac{1}{2}$ which is optimal (see later).

The Pentagon



Defender's strategy:
In the *n*-th step, choose uniformly between $A_{(n \mod 3)}$ and $B_{(n \mod 2)}$.

Intuition: Walk around A_i's, or B_i's, switch between these two walks with probability $1/2$ in every step.

Assume that Attacker attacks a node at an arbitrary time.

The probability of successful defence is $\frac{1}{2}$ which is optimal (see later).

No Defender's strategy with bounded memory can achieve $\frac{1}{2}$.

Roughly speaking, every optimal strategy gets eventually to the above pattern of watching modulo 2 and 3 which is impossible without time keeping.

Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

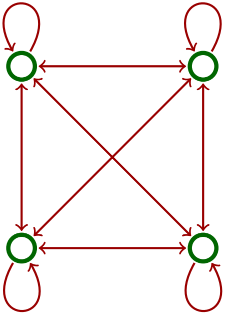
Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

If d divides $|U|$, then an optimal Defender's strategy is easy to construct:

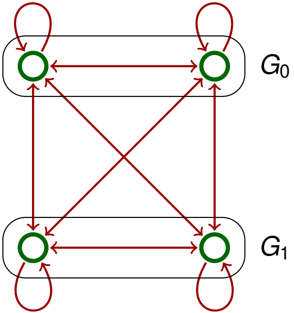


Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

If d divides $|U|$, then an optimal Defender's strategy is easy to construct:



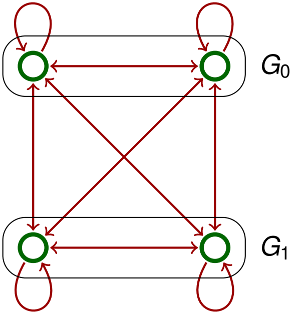
- Partition nodes of U to d groups $G_0, \ldots, G_{d-1}$ each of size $|U|/d$.
- In a node of G_i, move to a node of $G_{(i+1) \bmod d}$ chosen uniformly in random.

Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

If d divides $|U|$, then an optimal Defender's strategy is easy to construct:



- Partition nodes of U to d groups $G_0, \ldots, G_{d-1}$ each of size $|U|/d$.
- In a node of G_i, move to a node of $G_{(i+1) \bmod d}$ chosen uniformly in random.

The probability of successful defence against any attack is $d/|U|$.

Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

It cannot be improved:

$$\inf_{\pi} P[\sigma, \pi] \quad \leq \quad \frac{d}{|U|}$$

for every Defender's strategy σ.

Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

It cannot be improved:

$$\inf_{\pi} P[\sigma, \pi] \quad \leq \quad \frac{d}{|U|}$$

for every Defender's strategy σ.

Proof: Denote by $p^n(u)$ the probability of being in u after exactly n steps according to σ. Then

$$\sum_{n=1}^{d} p^n(u) \quad \geq \quad \inf_{\pi} P[\sigma, \pi]$$

Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

It cannot be improved:

$$\inf_{\pi} P[\sigma, \pi] \quad \leq \quad \frac{d}{|U|}$$

for every Defender's strategy σ.

Proof: Denote by $p^n(u)$ the probability of being in u after exactly n steps according to σ. Then

$$\sum_{u \in U} \sum_{n=1}^{d} p^n(u) \quad \geq \quad \sum_{u \in U} \inf_{\pi} P[\sigma, \pi] = |U| \inf_{\pi} P[\sigma, \pi]$$

Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

It cannot be improved:

$$\inf_{\pi} P[\sigma, \pi] \quad \leq \quad \frac{d}{|U|}$$

for every Defender's strategy σ.

Proof: Denote by $p^n(u)$ the probability of being in u after exactly n steps according to σ. Then

$$\sum_{n=1}^{d} \sum_{u \in U} p^n(u) = \sum_{u \in U} \sum_{n=1}^{d} p^n(u) \quad \geq \quad \sum_{u \in U} \inf_{\pi} P[\sigma, \pi] = |U| \inf_{\pi} P[\sigma, \pi]$$

Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

It cannot be improved:

$$\inf_{\pi} P[\sigma, \pi] \quad \leq \quad \frac{d}{|U|}$$

for every Defender's strategy σ.

Proof: Denote by $p^n(u)$ the probability of being in u after exactly n steps according to σ. Then

$$d = \sum_{n=1}^{d} 1 = \sum_{n=1}^{d} \sum_{u \in U} p^n(u) = \sum_{u \in U} \sum_{n=1}^{d} p^n(u) \quad \geq \quad \sum_{u \in U} \inf_{\pi} P[\sigma, \pi] = |U| \inf_{\pi} P[\sigma, \pi]$$

Complete Set of Arcs (Single Attack Length)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Assume that for all $u \in U$ we have $d(u) = d$ for a fixed $d \in \mathbb{N}$.

It cannot be improved:

$$\inf_{\pi} P[\sigma, \pi] \quad \leq \quad \frac{d}{|U|}$$

for every Defender's strategy σ.

Proof: Denote by $p^n(u)$ the probability of being in u after exactly n steps according to σ. Then

$$d = \sum_{n=1}^{d} 1 = \sum_{n=1}^{d} \sum_{u \in U} p^n(u) = \sum_{u \in U} \sum_{n=1}^{d} p^n(u) \quad \geq \quad \sum_{u \in U} \inf_{\pi} P[\sigma, \pi] = |U| \inf_{\pi} P[\sigma, \pi]$$

This upper bound is generic and holds for an arbitrary topology of the game.

Complete Set of Arcs (Multiple Attack Lengths)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Complete Set of Arcs (Multiple Attack Lengths)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Denote by $S(k)$ the number of nodes u with $d(u) = k$.

Complete Set of Arcs (Multiple Attack Lengths)

Assume that for every pair $u, v \in U$ there is an arc from u to v.

Denote by $S(k)$ the number of nodes u with $d(u) = k$.

Then the previous bound can be generalized: Let $d_1, \ldots, d_\ell$ be all attack lengths assigned to nodes

$$\inf_{\pi} P[\sigma, \pi] \quad \leq \quad \left(\frac{S(d_1)}{d_1} + \cdots + \frac{S(d_\ell)}{d_\ell} \right)^{-1}$$

Complete Set of Arcs (Multiple Attack Lengths)

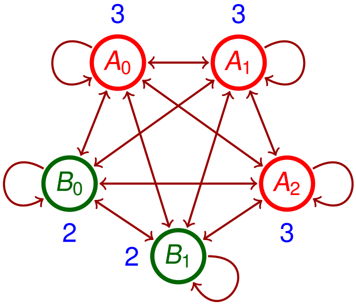
Assume that for every pair $u, v \in U$ there is an arc from u to v.

Denote by $S(k)$ the number of nodes u with $d(u) = k$.

Then the previous bound can be generalized: Let $d_1, \ldots, d_\ell$ be all attack lengths assigned to nodes

$$\inf_\pi P[\sigma, \pi] \quad \leq \quad \left(\frac{S(d_1)}{d_1} + \cdots + \frac{S(d_\ell)}{d_\ell} \right)^{-1}$$

For the pentagon we get $(2/2 + 3/3)^{-1} = 1/2$ which is the value of the proposed strategy.



Summary of Our Attempts

- We have found optimal strategies for complete graphs either
  - assuming that each attack length d divides the number of nodes $S(d)$ with the attack length d, or
  - assuming $d(u) = 2$ for all $u \in U$.

Summary of Our Attempts

- We have found optimal strategies for complete graphs either
  - assuming that each attack length d divides the number of nodes $S(d)$ with the attack length d, or
  - assuming $d(u) = 2$ for all $u \in U$.
- We conjecture a structure of strategies for other complete graphs (no proof yet).
- Arbitrary graphs??

Summary of Our Attempts

- We have found optimal strategies for complete graphs either
  - assuming that each attack length d divides the number of nodes $S(d)$ with the attack length d, or
  - assuming $d(u) = 2$ for all $u \in U$.
- We conjecture a structure of strategies for other complete graphs (no proof yet).
- Arbitrary graphs??

Does there always exist an optimal strategy of Defender?

Existence of Optimal Defender's Strategy

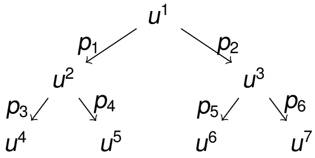
Theorem 2
There always exists Defender's strategy σ^* such that

$$\inf_{\pi} P[\sigma^*, \pi] = \sup_{\sigma} \inf_{\pi} P[\sigma, \pi]$$

Proof uses reduction to two-player *perfect-information* turn-based *stochastic* games with reachability objectives.

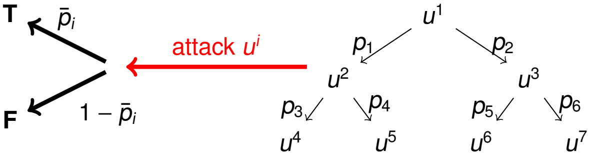
Existence of Optimal Defender's Strategy

For illustration, assume the attack length 2 everywhere and assume that each node has exactly two successors.



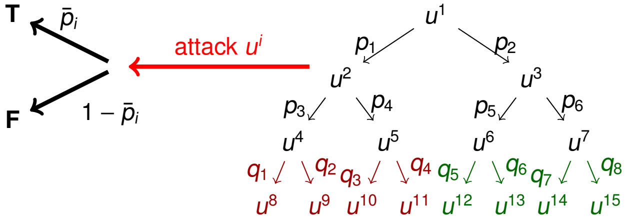
Existence of Optimal Defender's Strategy

For illustration, assume the attack length 2 everywhere and assume that each node has exactly two successors.



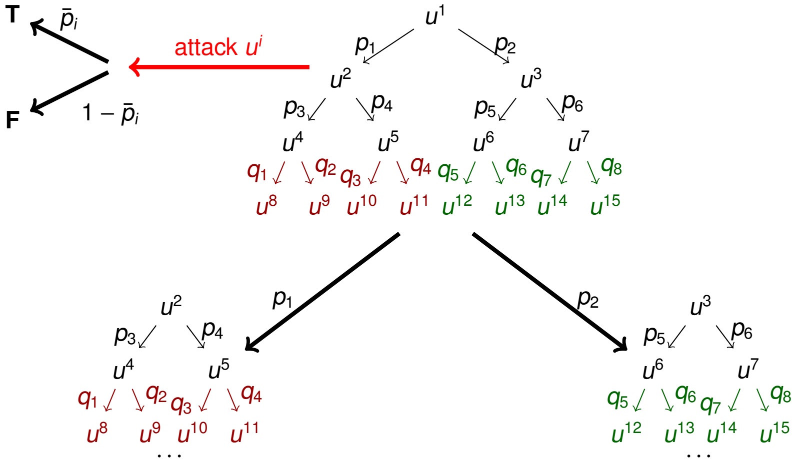
Existence of Optimal Defender's Strategy

For illustration, assume the attack length 2 everywhere and assume that each node has exactly two successors.



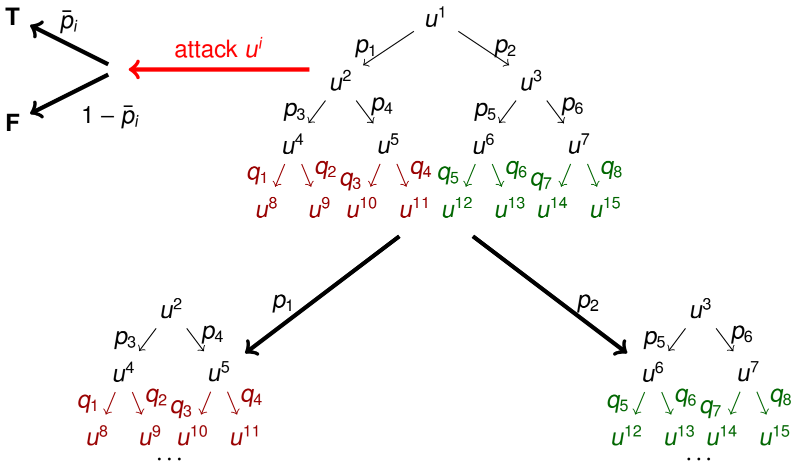
Existence of Optimal Defender's Strategy

For illustration, assume the attack length 2 everywhere and assume that each node has exactly two successors.



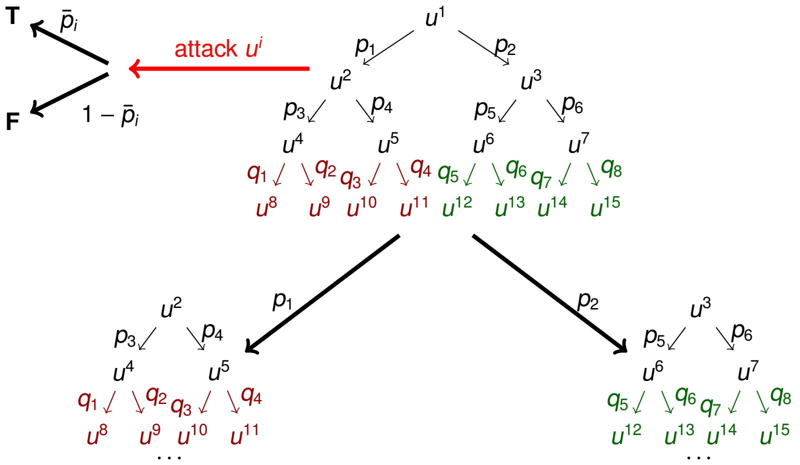
Existence of Optimal Defender's Strategy

For illustration, assume the attack length 2 everywhere and assume that each node has exactly two successors.



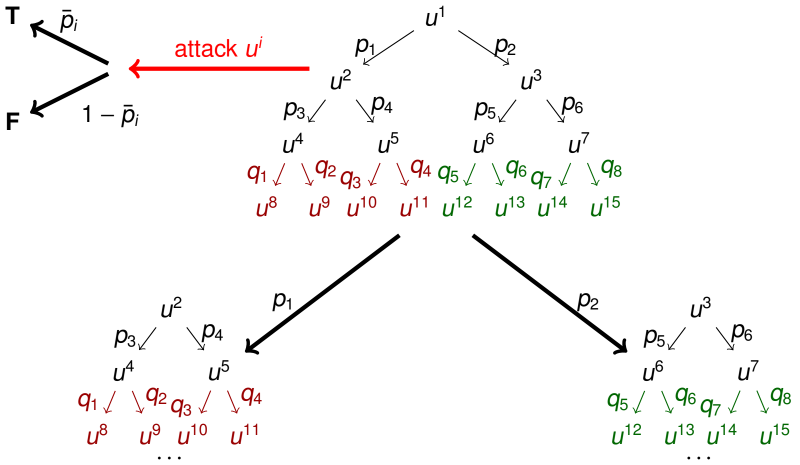
Attacker strives to reach **T**, defender avoids **T**.

Computing ε-Optimal Strategies



Round all numbers to a suitable precision – gives a finite state game.

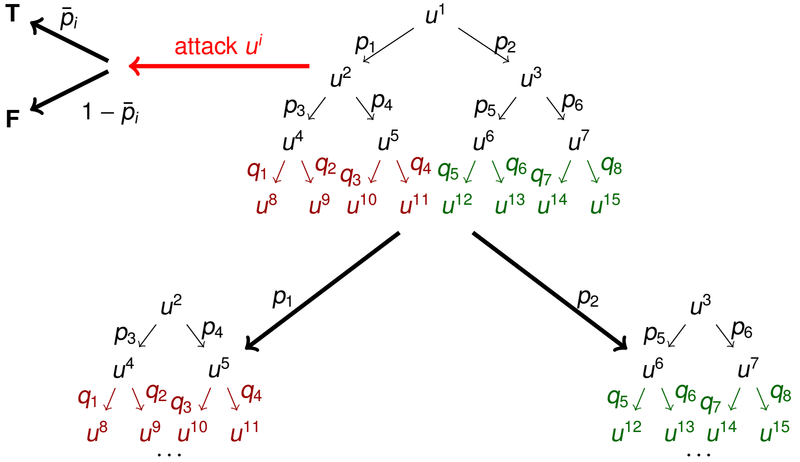
Computing ε-Optimal Strategies



Round all numbers to a suitable precision – gives a finite state game.

Solve the finite state game (naively in doubly exponential time).

Computing ε-Optimal Strategies



Round all numbers to a suitable precision – gives a finite state game.

Solve the finite state game (naively in doubly exponential time).

Exploiting symmetries gives *single exponential time* algorithm for computing ε-optimal strategies.

Conclusions

- We investigate existence and structure of optimal Defender's strategies in Stackelberg patrolling games.
- We have found optimal strategies in several special cases.
- We have constructed a generic algorithm for computing ε-optimal Defender's strategies in exponential time.

Conclusions

- We investigate existence and structure of optimal Defender's strategies in Stackelberg patrolling games.
- We have found optimal strategies in several special cases.
- We have constructed a generic algorithm for computing ε-optimal Defender's strategies in exponential time.

Future work:

- Extend our results to the huge amount of richer models of patrolling games considered in literature.
- Find efficient methods for solving patrolling games that have guarantees on performance (ε-optimality).
- Exploit decomposition techniques, e.g. based on compositionality of blind strategies.