m-Eternal Domination Number of Cactus Graphs

Václav Blažej, Matyáš Křišťan, Tomáš Valla

Faculty of Information Technology - Czech Technical University in Prague

September 12, 2019

m-Eternal Domination

- Game of two players, an attacker and a defender, played on a graph.

m-Eternal Domination

- Game of two players, an attacker and a defender, played on a graph.
- Given a graph, the defender places guards on some vertices.

m-Eternal Domination

- Game of two players, an attacker and a defender, played on a graph.
- Given a graph, the defender places guards on some vertices.
- In each turn, the attacker chooses one vertex to attack.

m-Eternal Domination

- Game of two players, an attacker and a defender, played on a graph.
- Given a graph, the defender places guards on some vertices.
- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.

m-Eternal Domination

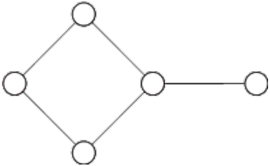
- Game of two players, an attacker and a defender, played on a graph.
- Given a graph, the defender places guards on some vertices.
- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.
- Each vertex can be occupied by at most one guard.

m-Eternal Domination

- Game of two players, an attacker and a defender, played on a graph.
- Given a graph, the defender places guards on some vertices.
- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.
- Each vertex can be occupied by at most one guard.
- **Question**: Given a graph G, what is the minimum number of guards required to defend it against any sequence of attacks?

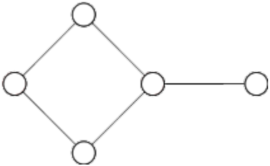
m-Eternal Domination

Input graph G:

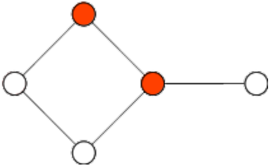


m-Eternal Domination

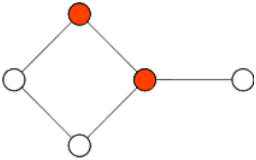
Input graph G:



Suppose we choose this initial configuration of guards (red):



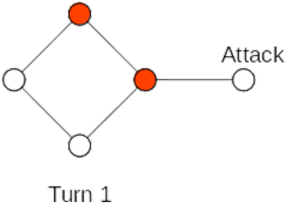
Example



Turn 1

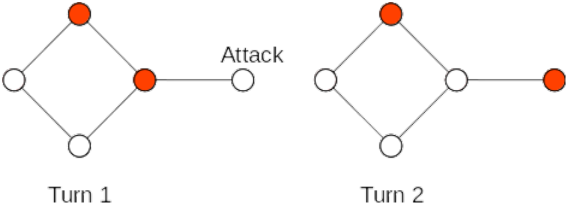
- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.

Example



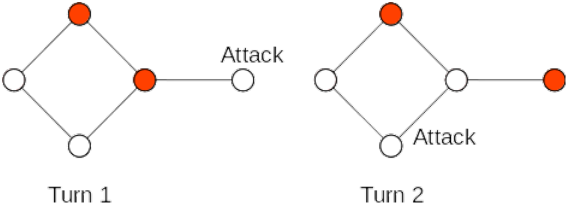
- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.

Example



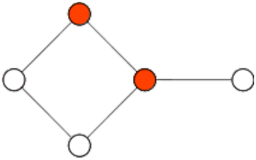
- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.

Example



- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.

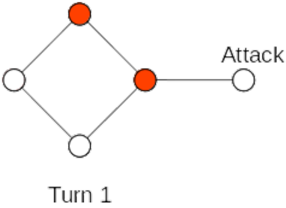
Example 2



Turn 1

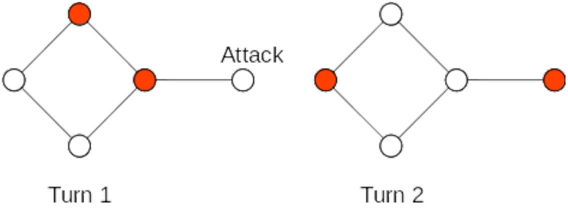
- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.

Example 2



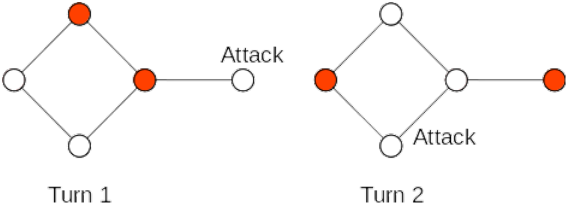
- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.

Example 2



- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.

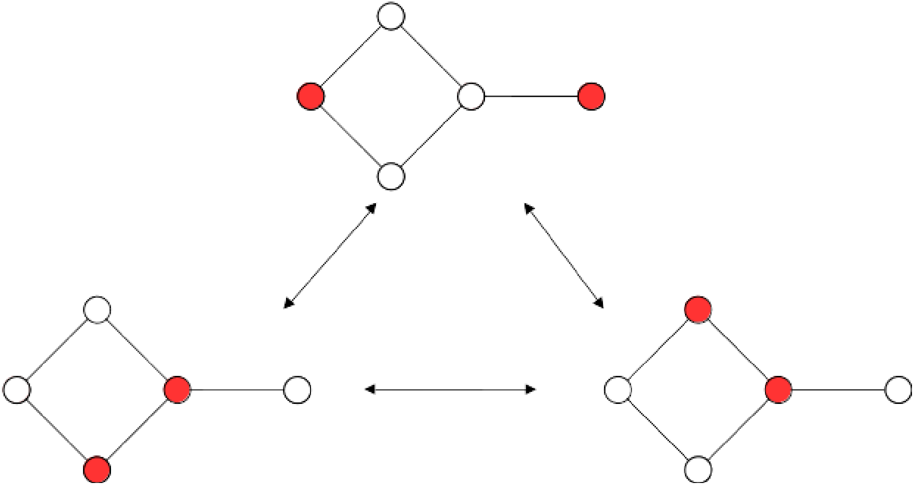
Example 2



- In each turn, the attacker chooses one vertex to attack.
- In response, the defender can move each of the guards into their respective neighborhoods. After the movements, the attacked vertex must be occupied. Otherwise, the defender loses.

Example

Three configurations of two guards are sufficient to defend against any sequence of attacks.



Known results

m-Eternal Domination - decision variant:

Input: Graph G, integer k
Output: Can k guards defend G against any possible sequence of attacks?

- Known to be NP-Hard
- Lies in EXPTIME
- Unknown whether lies in PSPACE

Known results

- Solved for cycles, complete graphs, complete bipartite graphs [Goddard et al. 2005]
- Linear algorithm for trees [Klostermeyer, MacGillivray, 2009]
- Linear algorithm for interval graphs [Rinemberg, Soulignac, 2019]

Let $\gamma_m^\infty(G)$ be the minimum number of guards required to defend G against any sequence of attacks.

Theorem (Henning, Klostermeyer and MacGillivray, 2017)

Let G be a connected graph with minimum degree $\delta(G) \geq 2$ which has $n \neq 4$ vertices. Then $\gamma_m^\infty(G) \leq \lfloor (n-1)/2 \rfloor$ and this bound is tight.

Known results

Theorem (Finbow, Messinger and van Bommel, 2015)

For $n \geq 2$,

$$\gamma_m^\infty(P_3 \square P_n) \leq \lceil 6n/7 \rceil + \begin{cases} 1 & \text{if } n \equiv 7, 8, 14 \text{ or } 15 \pmod{21} \\ 0 & \text{otherwise} \end{cases}$$

Theorem (van Bommel, van Bommel, 2016)

$$\lfloor \frac{6n+9}{5} \rfloor \leq \gamma_m^\infty(P_5 \square P_n) \leq \lfloor \frac{4n+4}{3} \rfloor$$

- More known results can be found in *Protecting a Graph with Mobile Guards* [Klostermeyer, Mynhardt, 2015]

Our results

Theorem

Let G be a Christmas cactus graph. Then there exists a linear-time algorithm which computes the minimum required number of guards to defend G.

Christmas cactus graph

Definition

Cactus graph is a connected graph in which every edge lies on at most one cycle.

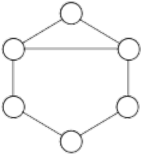
Christmas cactus graph

Definition

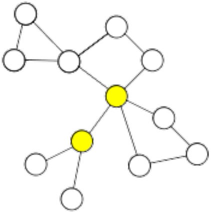
Cactus graph is a connected graph in which every edge lies on at most one cycle.

Definition

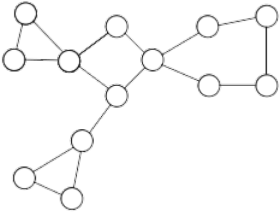
Christmas cactus graph is a cactus graph, in which removal of any vertex splits the graph into at most two connected components.



Not a cactus



Cactus
Not a Christmas cactus



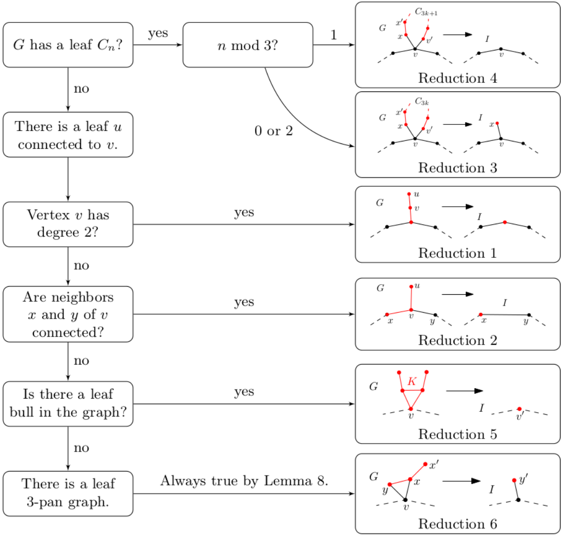
Christmas cactus

Algorithm

- Algorithm works as follows:
  - If G is a trivial case (cycle, K_2, K_1), output result.
  - Otherwise, perform one of possible reductions, which decrease γ_m^∞ by a constant known amount. The result of the reduction is always a smaller Christmas cactus graph.
  - Repeat with the reduced graph.

Overview of the reductions

If G is not trivial (cycle, K_2, K_1):



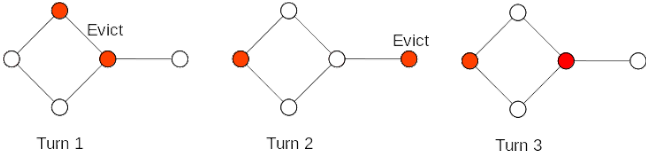
Idea of the proof

- Use two games, one for upper bound and the other for lower bound on the optimal number of guards in m-eternal domination.

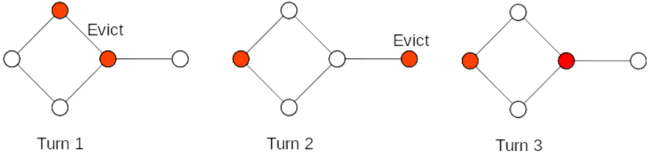
Idea of the proof

- Use two games, one for upper bound and the other for lower bound on the optimal number of guards in m-eternal domination.
- **Upper bound:** m-eternal domination with eviction

Idea of the proof

- Use two games, one for upper bound and the other for lower bound on the optimal number of guards in m-eternal domination.
- **Upper bound:** m-eternal domination with eviction
  - Instead of attacking, the attacker can pick an edge on a cycle or a vertex, which has to be evicted by guards.
  - 

Idea of the proof

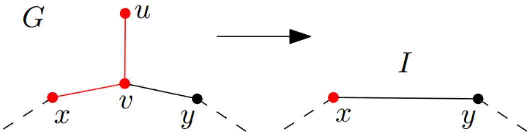
- Use two games, one for upper bound and the other for lower bound on the optimal number of guards in m-eternal domination.
- **Upper bound:** m-eternal domination with eviction
  - Instead of attacking, the attacker can pick an edge on a cycle or a vertex, which has to be evicted by guards.
  - 

- **Lower bound:** allow multiple guards on one vertex

Idea of the proof

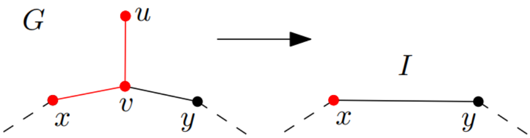
- The relationship for the optimal number of guards between these games is as follows:
- **m-eternal domination with multiple guards on one vertex**
  $\leq$
  m-eternal domination
  $\leq$
  m-eternal domination with eviction

Example of a reduction



- Upper bound
  - Assume an optimal strategy on I for m-eternal domination with eviction.
  - Extend it to G while adding one guard.
  - Show how to use the new guard to defend u and v.
  - Show how to evict edges $\{x, v\}$, $\{v, y\}$ and vertices u, v.
  - Show how to simulate a guard passing through $\{x, y\}$ in I, so the strategy is applicable in G.

Example of a reduction



- Lower bound
  - Assume an optimal strategy on G for m-eternal domination with multiple guards on one vertex.
  - Adapt the strategy of G for I.
  - Contract edges $\{u, v\}$ and $\{v, y\}$. Any guard moving along a contracted edge does not move.
  - Show that there is a guard which never leaves y and is not necessary and therefore can be removed.
  - The resulting strategy defends I and uses one less guard than in G.

Summary

- Linear algorithm for Christmas cactus graphs
- Upper bound on the optimal number of guards for cactus graphs based on a decomposition into Christmas cactus graphs.

Open problems

- More efficient (than EXPTIME) algorithm for graphs with treewidth 2.
- Algorithm parameterized by treewidth.
- Does the decision variant of the problem lie in PSPACE? Is it EXPTIME hard?
- Can we bound the number of required guard configurations in an optimal strategy?

The end

Thank you for your attention!