Spy Game on Graphs

Nathann Cohen\textsuperscript{1}  Nícolas A. Martins\textsuperscript{2}  Fionn Mc Inerney\textsuperscript{3}
Nicolas Nisse\textsuperscript{3}  Stéphane Pérennes\textsuperscript{3}  Rudini Sampaio\textsuperscript{2}

\textsuperscript{1}CNRS, Univ Paris Sud, LRI, Orsay, France
\textsuperscript{2}Universidade Federal do Ceará, Fortaleza, Brazil
\textsuperscript{3}Université Côte d’Azur, Inria, CNRS, I3S, France

Lyon, France, October 24, 2017

GAG Workshop
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

  Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

  Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

- Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)
- Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Pursuit-Evasion Games

- Mobile agents in a graph.
- Turn-by-turn with 2 players.
- Coordination for common goal, e.g.,

Cops and Robbers (capture) (Quilliot, 1978; Nowakowski, Winkler, 1983; Bonato, Nowakowski, 2011)

Eternal Domination (protection) (Goddard et al, 2005; Klostermeyer, MacGillivray, 2009)
Spy Game

Spy ($1^{st}$) vs guards ($2^{nd}$) in a graph $G$.

**Start**: Spy placed at a vertex. Then, guards placed.

**Turn-by-turn**: Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

**Goal**: Spy wants to be at least distance $d + 1$ from all guards.

Ex: $s = 2$ and $d = 1$. 
Spy (1\textsuperscript{st}) vs guards (2\textsuperscript{nd}) in a graph $G$.

\textbf{Start} : Spy placed at a vertex. Then, guards placed.

\textbf{Turn-by-turn} : Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

\textbf{Goal} : Spy wants to be at least distance $d + 1$ from all guards.

Ex : $s = 2$ and $d = 1$. 
Spy Game

Spy ($1^{st}$) vs guards ($2^{nd}$) in a graph $G$.

Start: Spy placed at a vertex. Then, guards placed.

Turn-by-turn: Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

Goal: Spy wants to be at least distance $d + 1$ from all guards.

Ex: $s = 2$ and $d = 1.$
Spy Game

Spy (1\textsuperscript{st}) vs guards (2\textsuperscript{nd}) in a graph $G$.

**Start**: Spy placed at a vertex. Then, guards placed.

**Turn-by-turn**: Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

**Goal**: Spy wants to be at least distance $d + 1$ from all guards.

Ex: $s = 2$ and $d = 1$. 
Spy Game

Spy ($1^{st}$) vs guards ($2^{nd}$) in a graph $G$.

**Start** : Spy placed at a vertex. Then, guards placed.

**Turn-by-turn** : Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

**Goal** : Spy wants to be at least distance $d + 1$ from all guards.

Ex : $s = 2$ and $d = 1$. 
Spy (1\textsuperscript{st}) vs guards (2\textsuperscript{nd}) in a graph $G$.

Start: Spy placed at a vertex. Then, guards placed.

Turn-by-turn: Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

Goal: Spy wants to be at least distance $d + 1$ from all guards.

Ex: $s = 2$ and $d = 1$. 
Spy (1\textsuperscript{st}) vs guards (2\textsuperscript{nd}) in a graph $G$.

\textbf{Start} : Spy placed at a vertex. Then, guards placed.

\textbf{Turn-by-turn} : Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

\textbf{Goal} : Spy wants to be at least distance $d + 1$ from all guards.

Ex : $s = 2$ and $d = 1$. 
Spy Game

Spy (1\textsuperscript{st}) vs guards (2\textsuperscript{nd}) in a graph $G$.

**Start**: Spy placed at a vertex. Then, guards placed.

**Turn-by-turn**: Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

**Goal**: Spy wants to be at least distance $d + 1$ from all guards.

Ex : $s = 2$ and $d = 1$. 
Spy Game

Spy (1\textsuperscript{st}) vs guards (2\textsuperscript{nd}) in a graph $G$.

Start: Spy placed at a vertex. Then, guards placed.

Turn-by-turn: Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

Goal: Spy wants to be at least distance $d + 1$ from all guards.

Ex: $s = 2$ and $d = 1$. 
Spy Game

Spy (1\textsuperscript{st}) vs guards (2\textsuperscript{nd}) in a graph $G$.

**Start**: Spy placed at a vertex. Then, guards placed.

**Turn-by-turn**: Spy traverses up to $s \geq 2$ edges. Guards traverse up to 1 edge.

**Goal**: Spy wants to be at least distance $d + 1$ from all guards.

Ex: $s = 2$ and $d = 1$. 
Guard Number: $gn_{s,d}(G)$

**Definition**

For all $s \geq 2$, $d \geq 0$ and a graph $G$, $gn_{s,d}(G)$ is the minimum number of guards guaranteed to win vs the spy.
Guard Number: $gn_{s,d}(G)$

**Definition**

For all $s \geq 2$, $d \geq 0$ and a graph $G$, $gn_{s,d}(G)$ is the minimum number of guards guaranteed to win vs the spy.

\[
\begin{align*}
gn_{2,1}(G) &= 2 \\
 gn_{s,1}(G) &\leq \gamma(G)
\end{align*}
\]
Our Results: Computing $gn$

**Complexity**
Calculating $gn_{s,d}$ is NP-hard in general.

**Tight bounds for paths**

\[ gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+q} \right\rceil \]  
where \( q = \left\lfloor \frac{2d}{s-1} \right\rfloor \).

**Almost tight bounds for cycles**

\[ \left\lceil \frac{n+2q}{2(d+q)+3} \right\rceil \leq gn_{s,d}(C_n) \leq \left\lceil \frac{n+2q}{2(d+q)+1} \right\rceil \]  
where \( q = \left\lfloor \frac{2d}{s-1} \right\rfloor \).

**Polynomial time Linear Program for trees**
Can calculate $gn_{s,d}(T)$ and a corresponding strategy in polynomial time.

**Grids**

\[ \exists \beta > 0, \text{ s.t. } \Omega(n^{1+\beta}) \leq gn_{s,d}(G_{n \times n}) \].
Related Work

- Cops vs robber (capture at a distance) (Bonato et al, 2010).

- Eternal Domination (Goddard et al, 2005).

\[ \gamma_m(G) = \frac{g(s = \infty, d = 0)}{n} \]

- How many cops needed in an $n \times n$ grid?

\[ \leq \lceil \frac{mn}{5} \rceil + O(m + n) \] (Lamprou et al, 2016).
Cops vs robber (capture at a distance) (Bonato et al, 2010).

Cops vs fast robber (Fomin et al, 2010).
Related Work

- Cops vs robber (capture at a distance) (Bonato et al, 2010).
- Cops vs fast robber (Fomin et al, 2010).
  - How many cops needed in an $n \times n$ grid?

\[ \gamma_m(G) = gn_{s, d}(G) \text{ when } s = \infty \text{ and } d = 0. \]
Related Work

- Cops vs robber (capture at a distance) (Bonato et al, 2010).
- Cops vs fast robber (Fomin et al, 2010).
  - How many cops needed in an $n \times n$ grid?
- Eternal Domination (Goddard et al, 2005).
Cops vs robber (capture at a distance) (Bonato et al, 2010).

Cops vs fast robber (Fomin et al, 2010).

How many cops needed in an $n \times n$ grid?

Eternal Domination (Goddard et al, 2005).

$$\gamma^m(m \times n \text{ grid}) \leq \lceil \frac{mn}{5} \rceil + O(m + n)$$ (Lamprou et al, 2016).
Cops vs robber (capture at a distance) (Bonato et al, 2010).

Cops vs fast robber (Fomin et al, 2010).

- How many cops needed in an $n \times n$ grid?

Eternal Domination (Goddard et al, 2005).

- $\gamma^m(m \times n \text{ grid}) \leq \left\lceil \frac{mn}{5} \right\rceil + O(m + n)$ (Lamprou et al, 2016).

- $\gamma^m(G) = gn_{s,d}(G)$ when $s = \infty$ and $d = 0$. 
Theorem
For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$g_{n_s,d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$
Paths: Lower bound

**Theorem**

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$

**Ex:** $s = 3$ and $d = 1$. 

```
```

Cohen, Martins, McInerney, Nisse, Pérennes, Sampaio  Spy Game on Graphs
Paths: Lower bound

Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d + 2 + \lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$. 

---

Cohen, Martins, McInerney, Nisse, Pérennes, Sampaio

Spy Game on Graphs
Paths : Lower bound

Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$g_{n, s, d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$

Ex : $s = 3$ and $d = 1$. 
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$g_{n_s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

Ex : $s = 3$ and $d = 1$. 
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$g_{n,s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$. 
**Path: Lower bound**

**Theorem**

For all \( s \geq 2, d \geq 0 \), and a path \( P_n \) on \( n \) vertices,

\[
g_{n,s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil
\]

**Ex:** \( s = 3 \) and \( d = 1 \).
Paths: Lower bound

**Theorem**

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$g_{n,s,d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$

**Ex:** $s = 3$ and $d = 1$. 
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$. 

\[ \begin{align*}
\end{align*} \]
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2 + \lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$. 
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$g_{n,s,d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$. 

\begin{center}
\begin{tikzpicture}
  \node зайда (1) at (0,0) {}; 
  \node (2) at (1,0) {}; 
  \node (3) at (2,0) {}; 
  \node (4) at (3,0) {}; 
  \node зайда (5) at (4,0) {}; 
  \node (6) at (5,0) {}; 
  \node (7) at (6,0) {}; 
  \node зайда (8) at (7,0) {}; 
  \node зайда (9) at (8,0) {}; 
  \node заядло (20) at (9,0) {}; 
  \node заядло (21) at (10,0) {}; 
  \node заядло (22) at (11,0) {}; 
  \node заядло (23) at (12,0) {}; 
  \node заядло (24) at (13,0) {w} 
\end{tikzpicture}
\end{center}
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$g_{n,s,d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$.

$$g_{n,3,1}(P_{10}) = 2$$
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$.

1 guard can protect subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
**Theorem**

For all \( s \geq 2 \), \( d \geq 0 \), and a path \( P_n \) on \( n \) vertices,

\[
gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil
\]

**Ex:** \( s = 3 \) and \( d = 1 \).

1 guard can protect subpath of \( 2d + 2 + \lfloor \frac{2d}{s-1} \rfloor \) vertices.
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$.

1 guard can protect subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
Paths : Upper bound

Theorem
For all \( s \geq 2, \ d \geq 0 \), and a path \( P_n \) on \( n \) vertices,
\[
gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil
\]

Ex : \( s = 3 \) and \( d = 1 \).

1 guard can protect subpath of \( 2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor \) vertices.
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$.

1 guard can protect subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
**Theorem**

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\lfloor \frac{2d}{s-1} \rfloor} \right\rceil$$

**Ex:** $s = 3$ and $d = 1$.

1 guard can protect subpath of $2d + 2 + \lfloor \frac{2d}{s-1} \rfloor$ vertices.
**Theorem**

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices, 

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

Ex : $s = 3$ and $d = 1$.

1 guard can protect subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
Paths: Upper bound

**Theorem**

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

**Ex**: $s = 3$ and $d = 1$.

1 guard can protect subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
Paths: Upper bound

**Theorem**

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

**Ex:** $s = 3$ and $d = 1$.

1 guard can protect subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
Paths : Upper bound

**Theorem**

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$g_{n,s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

**Ex**: $s = 3$ and $d = 1$.

1 guard can protect subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
Paths: Upper bound

Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

Ex: $s = 3$ and $d = 1$.

1 guard can protect subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

$$g_{n,s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

Each guard protects a subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
Theorem

For all $s \geq 2$, $d \geq 0$, and a path $P_n$ on $n$ vertices,

\[ gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil \]

Each guard protects a subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
Cycles: Upper Bound Case $2d < s - 1$

**Theorem**

For all $s \geq 2$, $d \geq 0$ s.t. $2d < s - 1$, and a cycle $C_n$ on $n$ vertices,

$$g_{n,s,d}(C_n) = \left\lceil \frac{n}{2d+3} \right\rceil.$$  

Ex: $s = 6$ and $d = 0$.

$$g_{n,6,0}(C_{12}) = 4$$
Cycles: Upper Bound Case $2d < s - 1$

**Theorem**

For all $s \geq 2$, $d \geq 0$ s.t. $2d < s - 1$, and a cycle $C_n$ on $n$ vertices,

$$gn_{s,d}(C_n) = \left\lceil \frac{n}{2d+3} \right\rceil.$$

**Ex:** $s = 6$ and $d = 0$.

$$gn_{6,0}(C_{12}) = 4$$
Cycles : Upper Bound Case $2d < s - 1$

**Theorem**

For all $s \geq 2$, $d \geq 0$ s.t. $2d < s - 1$, and a cycle $C_n$ on $n$ vertices,

$$gn_{s,d}(C_n) = \left\lceil \frac{n}{2d+3} \right\rceil.$$

Ex : $s = 6$ and $d = 0$.

$$gn_{6,0}(C_{12}) = 4$$
Cycles: Upper Bound Case $2d < s - 1$

**Theorem**

For all $s \geq 2$, $d \geq 0$ s.t. $2d < s - 1$, and a cycle $C_n$ on $n$ vertices,

$$gn_{s,d}(C_n) = \left\lceil \frac{n}{2d+3} \right\rceil.$$

Ex: $s = 6$ and $d = 0$.

$$gn_{6,0}(C_{12}) = 4$$
Cycles : Upper Bound Case $2d < s − 1$

Theorem

For all $s \geq 2$, $d \geq 0$ s.t. $2d < s − 1$, and a cycle $C_n$ on $n$ vertices,

$$gn_{s,d}(C_n) = \left\lceil \frac{n}{2d+3} \right\rceil.$$  

Ex : $s = 6$ and $d = 0$.

$$gn_{6,0}(C_{12}) = 4$$
Cycles : Upper Bound Case $2d < s - 1$

Theorem

For all $s \geq 2$, $d \geq 0$ s.t. $2d < s - 1$, and a cycle $C_n$ on $n$ vertices,

$$gn_{s,d}(C_n) = \left\lceil \frac{n}{2d+3} \right\rceil.$$

Ex : $s = 6$ and $d = 0$.

$$gn_{6,0}(C_{12}) = 4$$
Cycles : Upper Bound Case $2d < s - 1$

Theorem

For all $s \geq 2$, $d \geq 0$ s.t. $2d < s - 1$, and a cycle $C_n$ on $n$ vertices,

$$g_{n,s,d}(C_n) = \left\lceil \frac{n}{2d+3} \right\rceil.$$  

Ex : $s = 6$ and $d = 0$.

$$g_{n_{6,0}}(C_{12}) = 4$$
Cycles

**Theorem**

For all \( s \geq 2, \ d \geq 0 \) s.t. \( q = 0 \), and a cycle \( C_n \) on \( n \) vertices, 
\[
gn_{s,d}(C_n) = \left\lceil \frac{n}{2d+3} \right\rceil.
\]

**Theorem**

For all \( s \geq 2, \ d \geq 0 \) s.t. \( q \neq 0 \), and a cycle \( C_n \) on \( n \) vertices, 
\[
\left\lceil \frac{n+2q}{2(d+q)+3} \right\rceil \leq gn_{s,d}(C_n) \leq \left\lceil \frac{n+2q}{2(d+q)+1} \right\rceil.
\]

**Reminder:** \( q = \left\lfloor \frac{2d}{s-1} \right\rfloor \).
Trees are Harder

Paths: 1 guard per subpath of $2d + 2 + \left\lfloor \frac{2d}{s-1} \right\rfloor$ vertices.
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $g_{n_{2,1}}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $\text{gn}_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 

Cohen, Martins, Mc Inerney, Nisse, Pérennes, Sampaio
Spy Game on Graphs
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree \( T \) where \( s = 2 \), \( d = 1 \) and \( \text{gn}_{2,1}(T) = 4 \).
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 
Trees are Harder

Can’t always divide tree into subtrees protected by a certain number of guards.

Example of a tree $T$ where $s = 2$, $d = 1$ and $gn_{2,1}(T) = 4$. 

Cohen, Martins, Mc Inerney, Nisse, Pérennes, Sampaio
Spy Game on Graphs
Fractional Version of the Game

- Guards may be \textit{fractional} entities; movements rep. by flows.

- Unchanged for spy. Total fraction of guards distance \( \leq d \) from spy must be \( \geq 1 \).
Guards may be **fractional** entities; movements rep. by flows.

Unchanged for spy. Total fraction of guards distance $\leq d$ from spy must be $\geq 1$.

\[ g_{n_2,1}(C_6) = 2 \]

but

1.5 guards suffice.
Fractional Version of the Game

- Guards may be fractional entities; movements rep. by flows.
- Unchanged for spy. Total fraction of guards distance $\leq d$ from spy must be $\geq 1$.

$\text{gn}_{2,1}(C_6) = 2$ but 1.5 guards suffice.
Fractional Version of the Game

- Guards may be **fractional** entities; movements rep. by flows.
- Unchanged for spy. Total fraction of guards distance \( \leq d \) from spy must be \( \geq 1 \).

\[ s = 2, \quad d = 1. \]

\[ gn_{2,1}(C_6) = 2 \]

but

1.5 guards suffice.
Fractional Version of the Game

- Guards may be fractional entities; movements rep. by flows.

- Unchanged for spy. Total fraction of guards distance $\leq d$ from spy must be $\geq 1$.

- Linear program to compute optimal fractional strategy.

\[ s = 2, \ d = 1. \]

\[ gn_{2,1}(C_6) = 2 \]

but

1.5 guards suffice.
Guards may be **fractional** entities; movements rep. by flows.

Unchanged for spy. Total fraction of guards distance $\leq d$ from spy must be $\geq 1$.

**Linear program** to compute optimal fractional strategy.

Optimal fractional strategy $\Rightarrow$ optimal integral strategy in trees.

$s = 2, \quad d = 1.$

$\text{Cohen, Martins, Mc Inerney, Nisse, Pérennes, Sampaio}$

Spy Game on Graphs
Theorem: Can transform optimal fractional strategy into optimal integral strategy in polynomial time.

Fractional Conf.
Theorem: Can transform optimal fractional strategy into optimal integral strategy in polynomial time.
Theorem: Can transform optimal fractional strategy into optimal integral strategy in polynomial time.
**Theorem**: Can transform optimal fractional strategy into optimal integral strategy in polynomial time.

Fractional Conf.  Transition Phase

**Theorem**: Can transform optimal fractional strategy into optimal integral strategy in polynomial time.

Fractional Conf.  Transition Phase  Integral Conf.
**Theorem**: Can transform optimal fractional strategy into optimal integral strategy in polynomial time.

Fractional Conf.  Transition Phase  Integral Conf.

Tree’s protection and guards’ movements preserved.

Tree’s protection and guards’ movements preserved.

**Fractional Conf.**

**Integral Conf.**

rounding
Restricted Strategies

\[ f : V^k \times V \Rightarrow V^k \] (Unrestricted strategy)

\[ \omega : V \Rightarrow V^k \] (Restricted strategy)
Restricted Strategies

\[ f : V^k \times V \Rightarrow V^k \text{ (Unrestricted strategy)} \]

\[ \omega : V \Rightarrow V^k \text{ (Restricted strategy)} \]

- Guards’ positions depend only on position of spy.
- 1 Unique configuration for guards for each position of spy.
Restricted Strategies

\[ f : V^k \times V \Rightarrow V^k \text{ (Unrestricted strategy)} \]

\[ \omega : V \Rightarrow V^k \text{ (Restricted strategy)} \]

- Guards’ positions depend only on position of spy.
- 1 Unique configuration for guards for each position of spy.

**Theorem**

Optimal fractional strategy \( \Rightarrow \) optimal fractional restricted strategy in trees.
Restricted Strategies

\[ f : \mathcal{V}^k \times \mathcal{V} \Rightarrow \mathcal{V}^k \text{ (Unrestricted strategy)} \]

\[ \omega : \mathcal{V} \Rightarrow \mathcal{V}^k \text{ (Restricted strategy)} \]

- Guards’ positions depend only on position of spy.
- 1 Unique configuration for guards for each position of spy.

**Theorem**

Optimal fractional strategy \( \Rightarrow \) optimal fractional restricted strategy in trees.

Can calculate optimal restricted fractional strategies with Linear Program in polynomial time.
Restricted strategy : $\omega : V \Rightarrow V^k$

$\omega_{x,u}$ : quantity of guards on $u$ when spy is on $x$.

$f_{x,x',u,u'}$ : quantity of guards that go from $u$ to $u'$ when spy goes from $x$ to $x'$.
Restricted strategy: \( \omega : V \Rightarrow V^k \)

\( \omega_{x,u} \): quantity of guards on \( u \) when spy is on \( x \).

\( f_{x,x',u,u'} \): quantity of guards that go from \( u \) to \( u' \) when spy goes from \( x \) to \( x' \).

(1) Minimize \( \sum_{v \in V} \omega_{x_0,v} \)

Minimize number of guards.
Linear Program

Restricted strategy: \( \omega : V \Rightarrow V^k \)

\( \omega_{x,u} \): quantity of guards on \( u \) when spy is on \( x \).

\( f_{x,x',u,u'} \): quantity of guards that go from \( u \) to \( u' \) when spy goes from \( x \) to \( x' \).

\[
(2) \sum_{v \in N_d[x]} \omega_{x,v} \geq 1 \quad \forall x \in V
\]

Guarantees always at least 1 guard within distance \( d \) of spy.
Restricted strategy: \( \omega : V \Rightarrow V^k \)

\( \omega_{x,u} \) : quantity of guards on \( u \) when spy is on \( x \).

\( f_{x,x',u,u'} \) : quantity of guards that go from \( u \) to \( u' \) when spy goes from \( x \) to \( x' \).

\[
\begin{align*}
(3) \quad \sum_{u' \in N[u]} f_{x,x',u,u'} &= \omega_{x,u} & \forall u \in V, x' \in N_s[x] \\
(4) \quad \sum_{u' \in N[u]} f_{x,x',u',u} &= \omega_{x',u} & \forall u \in V, x' \in N_s[x]
\end{align*}
\]

Guarantees validity of moves of guards when spy moves.
Restricted strategy: $\omega : V \Rightarrow V^k$

$\omega_{x,u}$: quantity of guards on $u$ when spy is on $x$.

$f_{x,x',u,u'}$: quantity of guards that go from $u$ to $u'$ when spy goes from $x$ to $x'$.

\[
(3) \sum_{u' \in N[u]} f_{x,x',u,u'} = \omega_{x,u} \quad \forall u \in V, x' \in N_s[x]
\]

\[
(4) \sum_{u' \in N[u]} f_{x,x',u',u} = \omega_{x',u} \quad \forall u \in V, x' \in N_s[x]
\]

Guarantees validity of moves of guards when spy moves.

$O(n^4)$ real variables and constraints.
Main Result: $gn$ in Trees

**Theorem**

\[ \forall s > 1, \ d \geq 0 \text{ and all trees } T, \ gn_{s,d}(T) \text{ and a corresponding strategy can be calculated in polynomial time.} \]

**Idea of proof**: Linear Program can compute opt. frac. restr. strategy in polynomial time.

Run LP. From previous theorem, strategy is opt. frac.

Can transform opt. frac. into opt. int. in polynomial time.
Grids

**Theorem**

\[ \exists \beta > 0, \text{ s.t. } \forall s > 1, \ d \geq 0, \ \Omega(n^{1+\beta}) \leq gn_{s,d}(G_{n \times n}). \]

**Idea of proof**: Lower bound holds for fractional version.
Grids

**Theorem**

\[ \exists \beta > 0, \text{ s.t. } \forall s > 1, \ d \geq 0, \ \Omega(n^{1+\beta}) \leq gn_{s,d}(G_{n \times n}). \]

**Idea of proof**: Lower bound holds for fractional version.

Torus and grid have same order of number of guards.

**Theorem**

\[ \exists \alpha \geq \log(3/2) \approx 0.58, \text{ s.t. } \forall s > 1, \ d \geq 0, \ fgn_{s,d}(G_{n \times n}) \leq O(n^{2-\alpha}). \]

**Idea of proof**: Density function \[ \omega^*(v) = \frac{c}{(dist(v,v_0)+1)^{\log 3/2}} \] for a constant \( c > 0 \) satisfies LP.
Distribution of Guards in the Torus for an optimal symmetrical spy-positional strategy when $n = 100$, $m = 100$, $s = 2$ and $d = 1$
Further Work

- Determine $gn_{s,d}(G_{n \times n})$.

- Approximate $gn_{s,d}(G)$ in polynomial time in certain classes of graphs?

- Fractional approach applied to other combinatorial games.
Thanks!