Spy game: FPT-algorithm and results on graph products

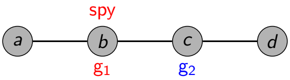
Rudini Sampaio
PARGO Research Group
Universidade Federal do Ceará (UFC)
Fortaleza, Brazil

co-authors
Eurinardo Costa (UFC, Russas, Brazil)
Nicolas Martins (UNILAB, Redenção, Brazil)

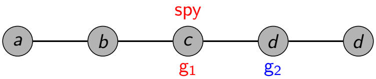
COCOON-2021, Tainan, Taiwan
Monday, October 25, 14h40 in Taiwan
03h40 in Brazil

(s, d)-Spy Game on Graphs

- (s, d): spy speed s and surveillance distance d
- **Instance:** Graph G and an integer k.
- **Players:** One spy and k guards occupy vertices of G.
- **Beginning:** The spy is placed first and then the guards.
- **Game:** Turn by turn, the spy may move along at most s edges and then each guard may move along one edge.
- **End:** Spy wins if she reaches a vertex at distance $> d$ from each guard. The guards win if a game configuration is played again.
- **Guard number $gn_{s,d}(G)$:** minimum k such that the guards have a winning strategy.



$gn_{2,0}(P_4) = 2$, $gn_{2,1}(P_4) = 1$



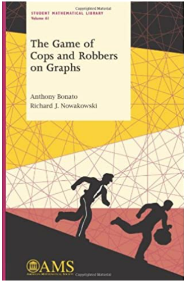
$gn_{2,0}(P_5) = 3$, $gn_{2,1}(P_5) = 1$

Spy Game: known results

- Introduced in [2016]. Exact value in paths/cycles.
- NP-hard for any $s \geq 2$ and $d \geq 0$ [2018].
- Polynomial time in trees [2020].

Other known pursuit games on graphs

- Cops and Robber [Nowakowski, Winkler'83]: cops are placed first and win if a cop occupies the same vertex of the robber. Well studied game, many variants, a lot of papers and 1 book.
- Similar for $s = 1$ if the guards were placed first in the Spy Game. Very different for $s \geq 2$.



Spy Game: known results

- Introduced in [2016]. Exact value in paths/cycles.
- NP-hard for any $s \geq 2$ and $d \geq 0$ [2018].
- Polynomial time in trees [2020].
- [2018]: Behaviour in grids? NP-hard in bipartite graphs?

Other known pursuit games on graphs

- Cops and Robber [Nowakowski, Winkler'83]: cops are placed first and win if a cop occupies the same vertex of the robber. Well studied game, many variants, a lot of papers and 1 book.
- Similar for $s = 1$ if the guards were placed first in the Spy Game. Very different for $s \geq 2$.
- Eternal Domination [Goddard et al.'2005]: Spy game with surveillance distance $s = \infty$ (or the diameter of the graph).

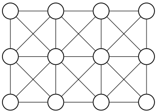
Spy Game: our results

- We begin the study of the spy game in grids and graph products.
- Strong product: $gn_{s,d}(G_1 \boxtimes G_2) \leq gn_{s_d}(G_1) \times gn_{s,d}(G_2)$.
- Strict upper bound: examples with King grids that match this upper bound and others for which the guard number is smaller.
- Cartesian/lexicographical products: this upper bound fails.
- Lexicographical products: Exact value for any distance $d \geq 2$.
- XP algorithm: The spy game decision problem is $O(n^{3k+2})$-time solvable for every speed $s \geq 2$ and distance $d \geq 0$.
- FPT algorithm: Spy Game is FPT on the P_4-fewness of the graph, solving a game on graphs for many graph classes.
- W[2]-hardness even in bipartite graphs when parameterized by the number k of guards, for every speed $s \geq 2$ and distance $d \geq 0$.

This hardness result generalizes the W[2]-hardness result of the spy game in general graphs [Cohen et al.'2018] and follows a similar (but significantly different) structure of the reduction from Set Cover in [2018]. However, the extension to bipartite graphs brings much more technical difficulties to the reduction, making this extension a relevant and non-trivial result.

Spy game on the strong product (and King grids)

The strong product $G_1 \boxtimes G_2$ has vertex set $V(G_1) \times V(G_2)$ and vertices (u_1, u_2) and (v_1, v_2) are adjacent iff (a) $u_1 = v_1$ and $u_2v_2 \in E(G_2)$, or (b) $u_2 = v_2$ and $u_1v_1 \in E(G_1)$, or (c) $u_1v_1 \in E(G_1)$ and $u_2v_2 \in E(G_2)$.
The King grid $\mathcal{K}_{n,m}$ is the strong product of two path graphs $P_n \boxtimes P_m$.



King Grid $4 \times 3 \;=\; P_4 \boxtimes P_3$

Theorem 1: Let $s \geq 2$ and $d \geq 0$.

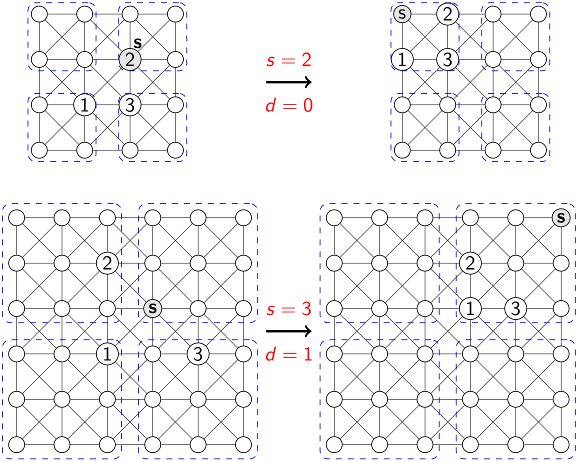
$$gn_{s,d}(G_1 \boxtimes G_2) \;\leq\; gn_{s,d}(G_1) \times gn_{s,d}(G_2).$$

Moreover, the equality holds if $gn_{s,d}(G_1) = 1$ or $gn_{s,d}(G_2) = 1$.

Proof sketch: Combining winning strategies in G_1 and G_2. For any guard h_1 in a vertex v_1 of G_1 and any guard h_2 in a vertex v_2 of G_2, consider a guard h_1h_2 in the vertex (v_1, v_2) of $G_1 \boxtimes G_2$.

Equality with $gn_{s,d}(G_1) = gn_{s,d}(G_2) = 2$

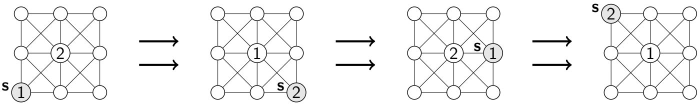
Lemma: For any $d \geq 0$, $s \geq d + 2$: $gn_{s,d}(P_{2d+4}) = 2$,
but $gn_{s,d}(P_{2d+4} \boxtimes P_{2d+4}) = 4$.



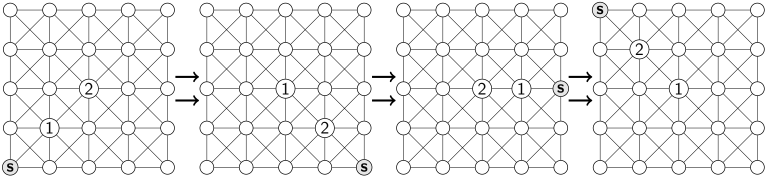
Strict UB with $gn_{s,d}(G_1) = gn_{s,d}(G_2) = 2$

Lemma: For any $d \geq 0$, $s \geq 2d + 2$: $gn_{s,d}(P_{2d+3}) = 2$,
but $gn_{s,d}(P_{2d+3} \boxtimes P_{2d+3}) \leq 2$.

Example: surveillance distance $d = 0$ and spy speed $s \geq 2$



Example: surveillance distance $d = 1$ and spy speed $s \geq 4$



$gn_{s,d}(G_1)$ and $gn_{s,d}(G_2)$ greater than 2

Lemma: Let $d \geq 0$, $2 \leq k \leq 2d+2$ and $s \geq (k-1)(2d+3)$. Then $gn_{s,d}(P_{k(2d+3)}) = k+1$ and $k^2 \leq gn_{s,d}(P_{k(2d+3)} \boxtimes P_{k(2d+3)}) \leq (k+1)^2$.

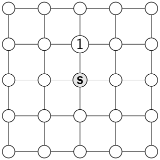
Proof: The vertex set of $P_{k(2d+3)} \boxtimes P_{k(2d+3)}$ can be partitioned into k^2 subsets of vertices which induces the King grid $P_{2d+3} \boxtimes P_{2d+3}$ each. If one of these subsets does not have a guard at some moment of the game, the spy can go to the vertex in the center of this subset and no guard can surveil the spy, which wins the game, since the distance from the center to the border is $d+1$ in the King grid $P_{2d+3} \boxtimes P_{2d+3}$. The spy can do this since her speed is at least $(k-1)(2d+3)$ (notice that the diameter is $k(2d+3)-1$ and the maximum distance between two centers of this subsets inducing the King grid $P_{2d+3} \boxtimes P_{2d+3}$ is at most $k(2d+3)-1-2(d+1) = (k-1)(2d+3)$). Thus, at least k^2 guards are necessary. Moreover, in [Cohen et al.'18], the exact value of $gn_{s,d}(P_n)$ was determined for any triple $(s \geq 2, d \geq 0, n \geq 2)$:

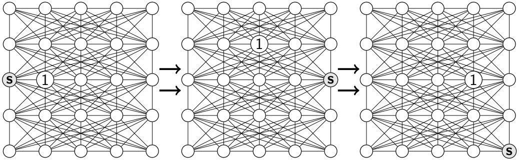
$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

With this, we have that $gn_{s,d}(P_{k(2d+3)}) = k+1$ for any $d \geq 0$, $2 \leq k \leq 2d+2$ and $s \geq (k-1)(2d+3) > 2d+1$. Then, from the upper bound, $(k+1)^2$ guards are sufficient and we are done. □

Cartesian and Lexicographical products

Lemma: $gn_{2,1}(P_5 \square P_5) > gn_{2,1}(P_5)^2$
$gn_{2,1}(P_5 \cdot P_5) > gn_{2,1}(P_5)^2$





Lexicographical products: exact value for $d \geq 2$

Theorem: Let $s \geq 2$, $d \geq 2$ and let G_1 and G_2 be two graphs. If G_1 has no isolated vertex, then:

$$gn_{s,d}(G_1 \cdot G_2) \quad = \quad gn_{s,d}(G_1).$$

Otherwise:

$$gn_{s,d}(G_1 \cdot G_2) \quad = \quad \max \{ \, gn_{s,d}(G_1), \; gn_{s,d}(G_2) \, \}.$$

XP-algorithm on the number of guards

Theorem: $k \geq 1$, $s \geq 2$ and $d \geq 0$. It is possible to decide in XP time $O(n^{3k+2})$ if the spy has a winning strategy against k guards in the (s, d)-spy game on G.

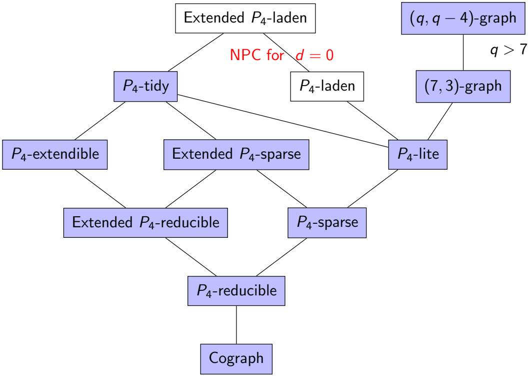
Proof: $2n^{k+1}$ game configurations.
Spy config: the spy is the next to move (spy/guards placed at vertices). Guard config: the guards are the next to move.
Spy winning configuration: spy at distance $> d$ from any guard.
Mark all spy winning configurations. **Repeat** the following until no more configurations are marked. Mark the guard configurations that only lead to marked spy configurations (in words, any guards' move will lead to a spy winning configuration). Mark the spy configurations that lead to at least one marked guard configuration (in words, there is a spy's move which leads to a spy winning configuration).

If there is a vertex u such that any spy config with the spy in u is marked, then the spy has a winning strategy (by occupying vertex u first). Otherwise, the guards have a winning strategy. □

FPT algorithm on the P_4-fewness

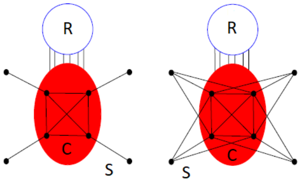
G is $(q, q-4)$ for $q \geq 4$ if any subset of $\leq q$ vertices induces $\leq q-4$ distinct P_4's. Cographs are $(4, 0)$ and P_4-sparse are $(5, 1)$-graphs. The **P_4-fewness** $q(G)$ is the min $q \geq 4$ s.t. G is a $(q, q-4)$-graph.

They are called "graphs with few P_4's" and are on the top of a very known hierarchy of graph classes. They also have a nice recursive decomposition based on unions, joins, spiders and small separable p-components.



FPT algorithm on the P_4-fewness

- Union $G = G_1 \cup G_2$: No edge between G_1 and G_2.
- Join $G = G_1 + G_2$: All edges between G_1 and G_2.



G is a **spider** if it has a partition (R, C, S) s.t.:

- $C = \{c_1, \ldots, c_k\}$ is a clique ($k \geq 2$);
- $S = \{s_1, \ldots, s_k\}$ is an independent set;
- All edges between R and C and no edge between R and S;
- **Thin spider**: s_i is adjacent to c_j if and only if $i = j$;
- **Thick spider**: s_i is adjacent to c_j if and only if $i \neq j$

FPT algorithm on the P_4-fewness

Lemma (Unions and Joins): $s \geq 2$ and $d \geq 0$:

$$gn_{s,d}(G_1 \cup G_2) \;=\; \max\left\{gn_{s,d}(G_1),\; gn_{s,d}(G_2)\right\}.$$

$$gn_{s,d}(G_1 + G_2) \;=\; \begin{cases} 1, & \text{if } d \geq 1 \text{ or } G_1 \text{ and } G_2 \text{ are complete,} \\ 2, & \text{otherwise.} \end{cases}$$

Lemma (spider (R, C, S)): If $d \geq 1$, then $gn_{s,d}(G) = 1$.
Moreover, with $s \geq 2$, if G is a thin spider, then

$$gn_{s,0}(G) \;=\; \begin{cases} |C|, & \text{if } R = \emptyset, \\ |C| + 1, & \text{otherwise,} \end{cases}$$

and, if G is a thick spider, then

$$gn_{s,0}(G) \;=\; \begin{cases} 2, & \text{if } R = \emptyset, \\ 3, & \text{otherwise,} \end{cases}$$

FPT algorithm on the P_4-fewness

Lemma (p-connected component): Let G with a separable p-component H with $|V(H)| < q$ and bipartition (H_1, H_2) of H s.t. any vertex of $G - H$ is adjacent to every vertex of H_1 and non-adjacent to every vertex of H_2. Then

$$gn_{s,d}(G) = gn_{s,d}(G_R) \leq q,$$

where G_R is the (reduced) graph obtained from G by replacing $G - H$ by two adjacent or non-adjacent vertices, depending whether $G - H$ is complete or not, respectively.

$(q, q-4)$-graphs and P_4-tidy graphs

- Decomposition theorems in terms of unions, joins, quasi-spiders and separable p-components
- Compute $gn_{s,d}(G)$ with the XP-algorithm (for fixed q).

FPT algorithm on the P_4-fewness

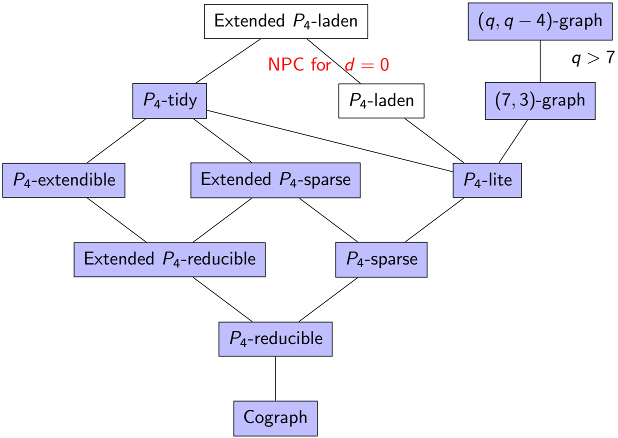
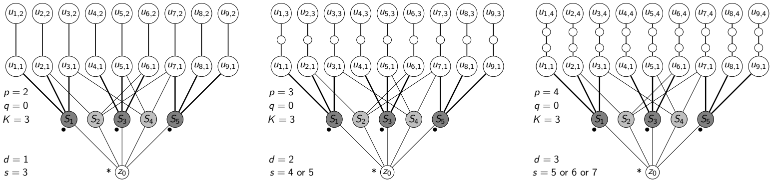


Figure: Known hierarchy of graphs with few P_4's

W[2]-hardness in bipartite graphs

- **Case 1:** $s > 2d + 2$
- **Case 2:** $s = 2d + 2$
- **Case 3:** $d + 1 < s < 2d + 2$
- **Case 4:** $s \leq d + 1$ and $s < 2(r + 1)$
- **Case 5:** $s \leq d + 1$ and $s = 2(r + 1)$
- **Case 6:** $s \leq d + 1$ and $s > 2(r + 1)$

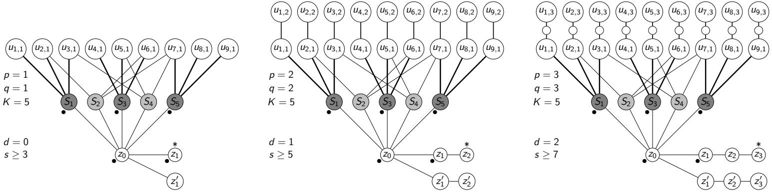
where $r = d \mod (s - 1)$ is the remainder of the division of d by $s - 1$.



W[2]-hardness in bipartite graphs

- **Case 1:** $s > 2d + 2$
- **Case 2:** $s = 2d + 2$
- **Case 3:** $d + 1 < s < 2d + 2$
- **Case 4:** $s \leq d + 1$ and $s < 2(r + 1)$
- **Case 5:** $s \leq d + 1$ and $s = 2(r + 1)$
- **Case 6:** $s \leq d + 1$ and $s > 2(r + 1)$

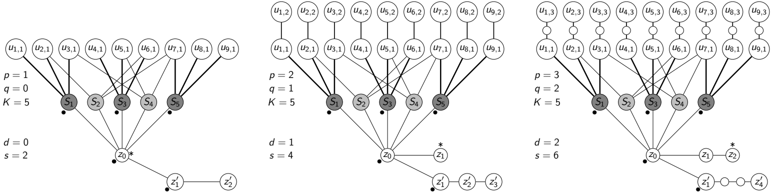
where $r = d \mod (s - 1)$ is the remainder of the division of d by $s - 1$.



W[2]-hardness in bipartite graphs

- **Case 1:** $s > 2d + 2$
- **Case 2:** $s = 2d + 2$
- **Case 3:** $d + 1 < s < 2d + 2$
- **Case 4:** $s \leq d + 1$ and $s < 2(r + 1)$
- **Case 5:** $s \leq d + 1$ and $s = 2(r + 1)$
- **Case 6:** $s \leq d + 1$ and $s > 2(r + 1)$

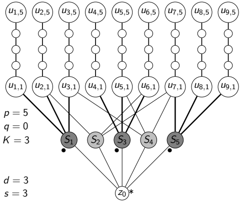
where $r = d \mod (s - 1)$ is the remainder of the division of d by $s - 1$.

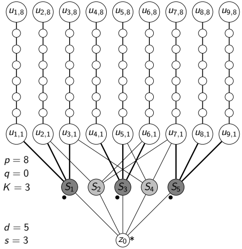


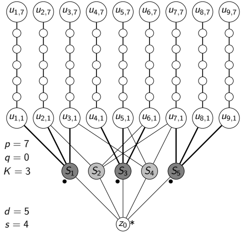
W[2]-hardness in bipartite graphs

- **Case 1:** $s > 2d + 2$
- **Case 2:** $s = 2d + 2$
- **Case 3:** $d + 1 < s < 2d + 2$
- **Case 4:** $s \leq d + 1$ and $s < 2(r + 1)$
- **Case 5:** $s \leq d + 1$ and $s = 2(r + 1)$
- **Case 6:** $s \leq d + 1$ and $s > 2(r + 1)$

where $r = d \mod (s - 1)$ is the remainder of the division of d by $s - 1$.



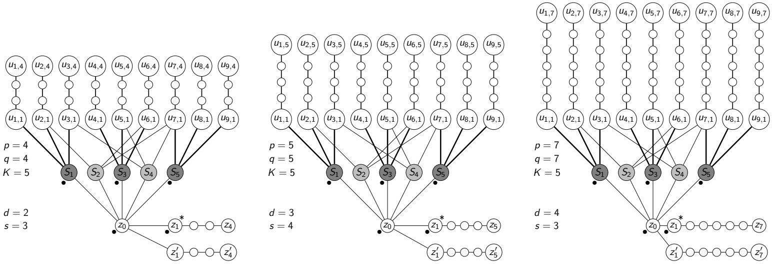




W[2]-hardness in bipartite graphs

- **Case 1:** $s > 2d + 2$
- **Case 2:** $s = 2d + 2$
- **Case 3:** $d + 1 < s < 2d + 2$
- **Case 4:** $s \leq d + 1$ and $s < 2(r + 1)$
- **Case 5:** $s \leq d + 1$ and $s = 2(r + 1)$
- **Case 6:** $s \leq d + 1$ and $s > 2(r + 1)$

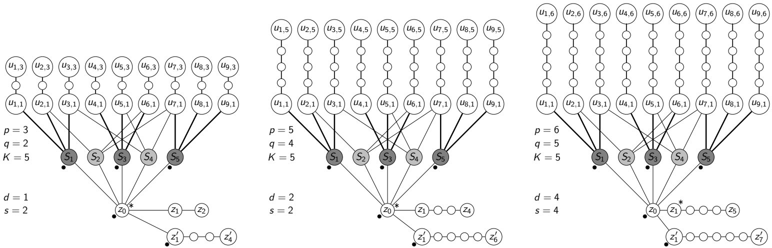
where $r = d \mod (s-1)$ is the remainder of the division of d by $s - 1$.



W[2]-hardness in bipartite graphs

- **Case 1:** $s > 2d + 2$
- **Case 2:** $s = 2d + 2$
- **Case 3:** $d + 1 < s < 2d + 2$
- **Case 4:** $s \leq d + 1$ and $s < 2(r + 1)$
- **Case 5:** $s \leq d + 1$ and $s = 2(r + 1)$
- **Case 6:** $s \leq d + 1$ and $s > 2(r + 1)$

where $r = d \mod (s - 1)$ is the remainder of the division of d by $s - 1$.



The END !!

THANK YOU !!

Spy game: FPT-algorithm and results on graph products

Rudini Sampaio
PARGO Research Group
Universidade Federal do Ceará (UFC)
Fortaleza, Brazil

co-authors
Eurinardo Costa (UFC, Russas, Brazil)
Nicolas Martins (UNILAB, Redenção, Brazil)

COCOON-2021, Tainan, Taiwan
Monday, October 25, 14h40 in Taiwan
03h40 in Brazil