Cops and Robber games and applications

Nicolas Nisse

Inria, France
Univ. Nice-Sophia Antipolis, I3S, CNRS, Sophia Antipolis, France

AlDyNet Workshop on Algorithms and Randomness

Santiago, November 21th, 2013

Graph structures and algorithmic

Problems arising from telecommunication networks

$\Rightarrow$ NP-hard, difficult to approximate

$\Rightarrow$ Polynomial but instances are huge (cf. David's talk)

Main approach: use networks (graphs) specificities/structures

- Real networks are specific $\Rightarrow$ algorithms must be specified
- Problems tractable in particular graph classes

  e.g., planar, bounded treewidth, preferencial attachment, etc.

  $\Rightarrow$ Fixed Parameter Tractable (FPT) algorithms, graph decompositions

Main tool: Pursuit-evasion games

- Models for studying several practical problems
- Offer new approaches for several structural graph properties
- Fun and intriguing questions

Graph structures and algorithmic

Problems arising from telecommunication networks

⇒ NP-hard, difficult to approximate
⇒ Polynomial but instances are huge (cf. David's talk)

Main approach: use networks (graphs) specificities/structures

- Real networks are specific ⇒ algorithms must be specified
- Problems tractable in particular graph classes

  e.g., planar, bounded treewidth, preferencial attachment, etc.

  ⇒ Fixed Parameter Tractable (FPT) algorithms, graph decompositions

Main tool: Pursuit-evasion games

- Models for studying several practical problems
- Offer new approaches for several structural graph properties
- Fun and intriguing questions

Pursuit-Evasion games

2-Player games on a graph

e.g., "Capture" an intruder in a network

Team of Cops/searchers (Player 1) *vs.* Robber/fugitive (Player 2)

Combinatorial Problem:

Minimizing some graph parameter

e.g., number of searchers to capture the fugitive.

Algorithmic Problem:

Computing strategy (sequence of moves) ensuring a Player to win

e.g., ensuring the searchers to capture the fugitive.

In this talk: 2 or 3 examples

definition/few results and applications/open problems

Outline

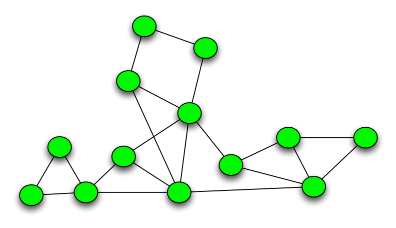
1. Graph Searching games
2. Cops and Robber games
3. Surveillance games

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network

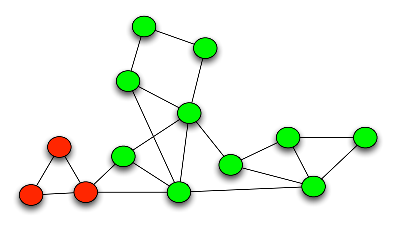


Initially, the whole graph is contaminated (fugitive may be anywhere)

Invisible Graph Searching [Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Cops are sequentially placed and removed from nodes...
Cops are sequentially placed and removed from nodes...

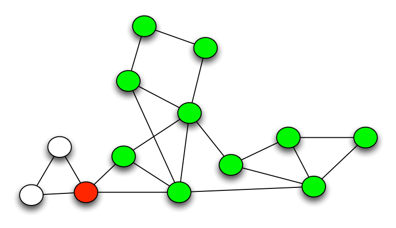
...clearing some nodes (white nodes)

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Cops are sequentially placed and removed from nodes...

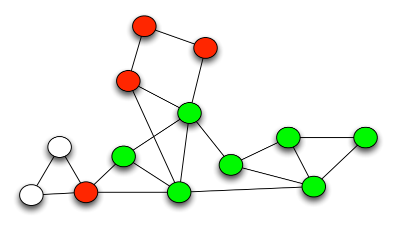
...clearing some nodes (white nodes)

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Cops are sequentially placed and removed from nodes...

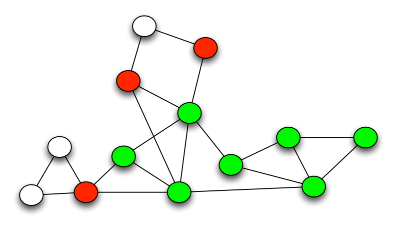
...clearing some nodes (white nodes)

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Cops are sequentially placed and removed from nodes...

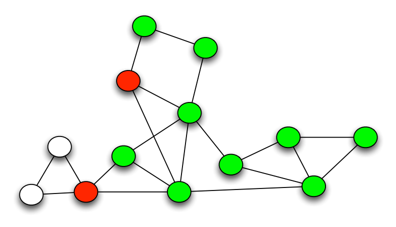
...clearing some nodes (white nodes)

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



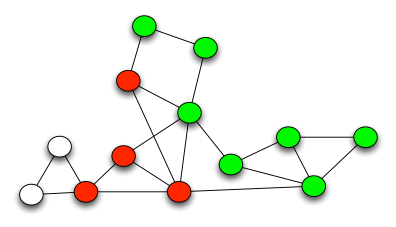
Recontamination may occur...

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Recontamination may occur...

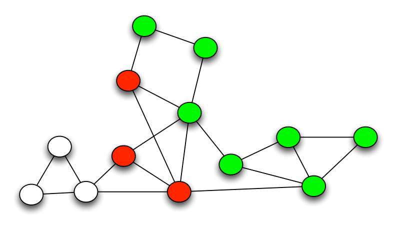
... but the strategy goes on

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



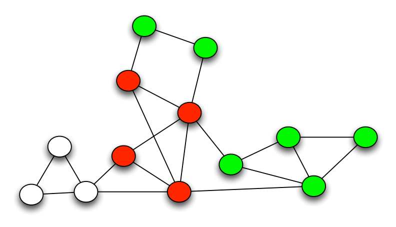
Recontamination may occur...

... but the strategy goes on

Invisible Graph Searching [Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Recontamination may occur...

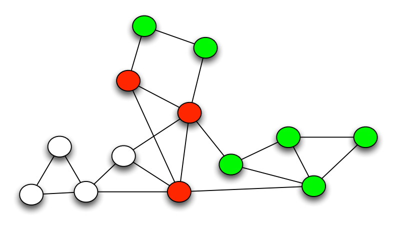
... but the strategy goes on

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



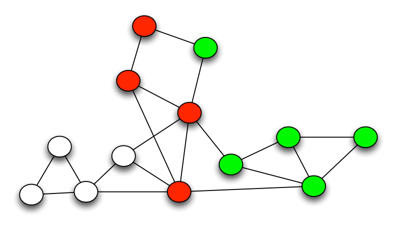
Recontamination may occur...

... but the strategy goes on

Invisible Graph Searching [Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Recontamination may occur...

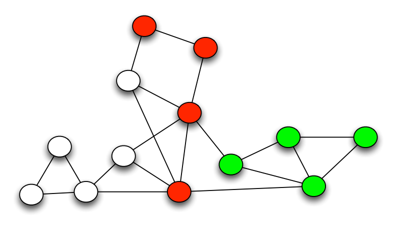
... but the strategy goes on

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Recontamination may occur...

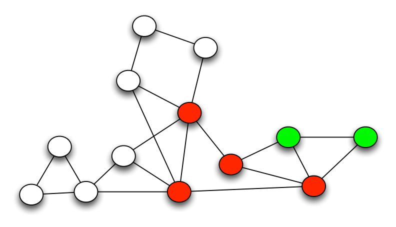
... but the strategy goes on

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Recontamination may occur...

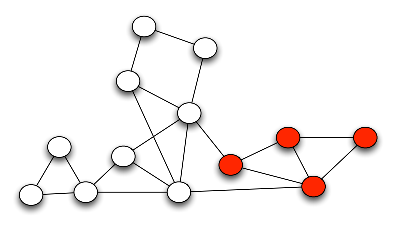
... but the strategy goes on

Invisible Graph Searching

[Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network

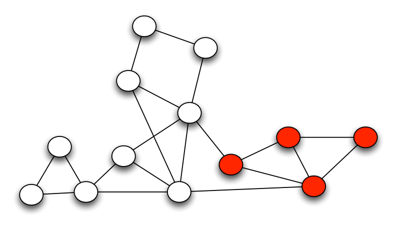


Graph G cleared with 4 cops (best possible, you can try) *search number*$(G) = 4$

Invisible Graph Searching [Breish'67, Parsons'76]

- invisible fugitive moves arbitrary fast, at any time, while not crossing cops
- cops can be placed or removed and must capture the fugitive

⇔ cops must clear a contaminated network



Computing the search number of graphs

NP-complete in planar graphs [Monien, Sudborough'88], chordal graphs [Gustedt'93], etc.

Linear in trees [Skodinis'03], Poly in circular-arc graphs [Todinca, Suchan'07], etc.

Invisible Graph Searching

Applications 1/3

Obviously: coordination of mobile autonomous agents

Drones tracking some target :([Guibas *et al*'99], Robots clearing a nuclear plant...

Connected GS: cleared area must be connected

$\forall G$, connected-sn$(G) \leq 2\ sn(G)$ [Dereniowski'12]

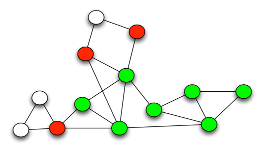
- Computing connected-sn(G) in NP?
- Is it FPT?

  i.e., can $csn(G) \leq k$ be decided in time $f(k) \cdot |G|^c$?

Distributed Algorithms:

$\approx$ autonomous robots must clear an unknown environment

[Flocchini *et al*'05,Ilcinkas,Soguet,N.'09,Angelo,Stephano,Navarra,N.,Suchan'13]...



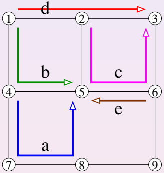
Example of non-connected step

Invisible Graph Searching

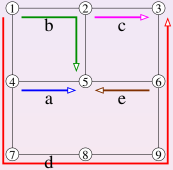
Applications 2/3

More surprising: Routing reconfiguration in WDM networks

Switching routes of requests, one by one, disturbing the traffic as few as possible



initial routing



final routing

Invisible Graph Searching

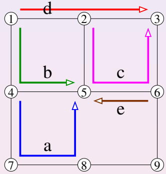
Applications 2/3

More surprising: Routing reconfiguration in WDM networks

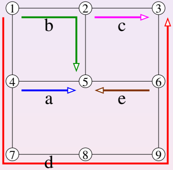
Switching routes of requests, one by one, disturbing the traffic as few as possible

Scheduling problem can be modeled as GS problem on the dependency digraph

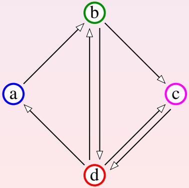
[Coudert,Sereni'11,Cohen *et al.*'11,Coudert,Huc,Mazauric'12]...



initial routing



final routing



Dependancy Digraph

- one vertex per connection with different routes in $\mathcal{I}$ and $\mathcal{F}$
- arc from u to v if ressources needed by u in $\mathcal{F}$ are used by v in $\mathcal{I}$

Graph Searching

Applications 3/3

Main(?) interest: Link with Graph Decompositions

Algorithmic applications: many hard problems are "easy" in bounded tw graph [Courcelle'90, Cygan *et al.*'11]...
Application to compact routing [Kosowski, Li, N. Suchan'12]

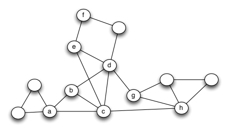
A graph and a path-decomposition
path decomposition ⇔ strategy for cops
$sn(G) = pw(G) + 1$ [Bienstock,Seymour'91]

Graph Searching

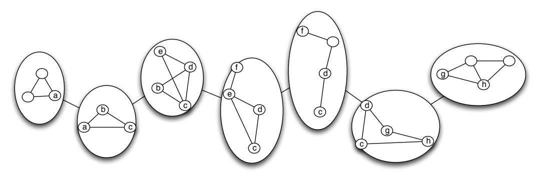
Applications 3/3

Main(?) interest: Link with Graph Decompositions

Algorithmic applications: many hard problems are "easy" in bounded tw graph [Courcelle'90, Cygan *et al.*'11]...
Application to compact routing [Kosowski, Li, N. Suchan'12]



A graph and a path-decomposition
path decomposition ⇔ strategy for cops
$sn(G) = pw(G) + 1$ [Bienstock,Seymour'91]

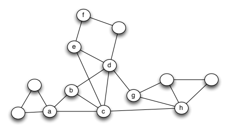


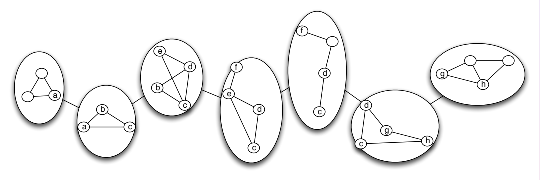
Graph Searching

Applications 3/3

Main(?) interest: Link with Graph Decompositions

Algorithmic applications: many hard problems are "easy" in bounded tw graph [Courcelle'90, Cygan *et al.*'11]...
Application to compact routing [Kosowski, Li, N. Suchan'12]





A graph and a path-decomposition

path decomposition ⇔ strategy for cops

$sn(G) = pw(G) + 1$ [Bienstock,Seymour'91]

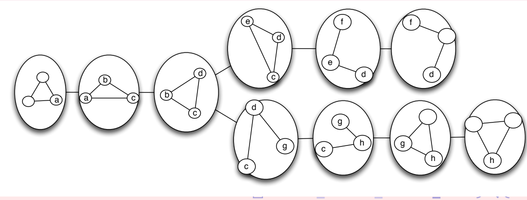
---

tree decomp. ⇔ strategy for cops vs. visible fugitive

visible-sn(G)=tw(G)+1 [Seymour,Thomas'93]

study of GS led to new results on treewidth

- duality [Seymour,Thomas'93, Amini,Mazoit,N.,Thomassé'09]...
- directed graphs [Adler'07,Ganian *et al.*'10]
  ⇒ **lot of work remains to do**



Outline

1. Graph Searching games
2. Cops and Robber games
3. Surveillance games

Cops & robber games

[Nowakowski and Winkler; Quilliot, 83]

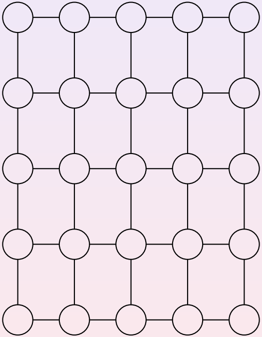
Initialization:

1. $\mathcal{C}$ places the cops;
2. $\mathcal{R}$ places the robber.

Step-by-step:

- each cop traverses at most 1 edge;
- the robber traverses at most 1 edge.

Robber captured:
A cop occupies the same vertex as the robber.



Cops & robber games

[Nowakowski and Winkler; Quilliot, 83]

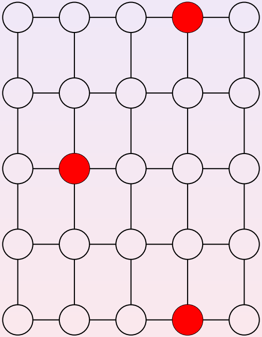
Initialization:

1. $\mathcal{C}$ places the cops;
2. $\mathcal{R}$ places the robber.

Step-by-step:

- each cop traverses at most 1 edge;
- the robber traverses at most 1 edge.

Robber captured:
A cop occupies the same vertex as the robber.



Cops & robber games

[Nowakowski and Winkler; Quilliot, 83]

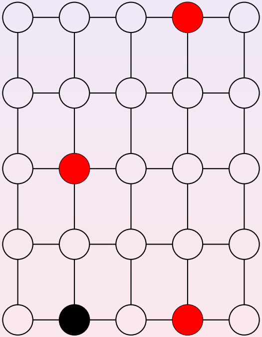
Initialization:

1. $\mathcal{C}$ places the cops;
2. $\mathcal{R}$ places the robber.

Step-by-step:

- each cop traverses at most 1 edge;
- the robber traverses at most 1 edge.

Robber captured:
A cop occupies the same vertex as the robber.



Cops & robber games

[Nowakowski and Winkler; Quilliot, 83]

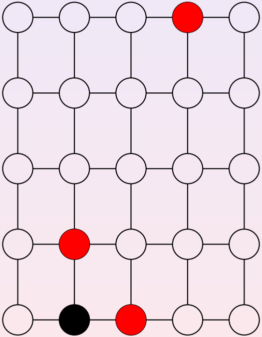
Initialization:

1. $\mathcal{C}$ places the cops;
2. $\mathcal{R}$ places the robber.

Step-by-step:

- each cop traverses at most 1 edge;
- the robber traverses at most 1 edge.

Robber captured:
A cop occupies the same vertex as the robber.



Cops & robber games

[Nowakowski and Winkler; Quilliot, 83]

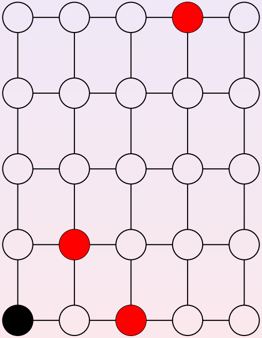
Initialization:

1. $\mathcal{C}$ places the cops;
2. $\mathcal{R}$ places the robber.

Step-by-step:

- each cop traverses at most 1 edge;
- the robber traverses at most 1 edge.

Robber captured:
A cop occupies the same vertex as the robber.



Cops & robber games

[Nowakowski and Winkler; Quilliot, 83]

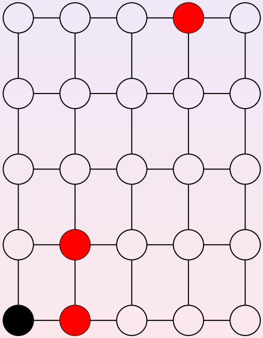
Initialization:

1. $\mathcal{C}$ places the cops;
2. $\mathcal{R}$ places the robber.

Step-by-step:

- each cop traverses at most 1 edge;
- the robber traverses at most 1 edge.

Robber captured:
A cop occupies the same vertex as the robber.



Cops & robber games

[Nowakowski and Winkler; Quilliot, 83]

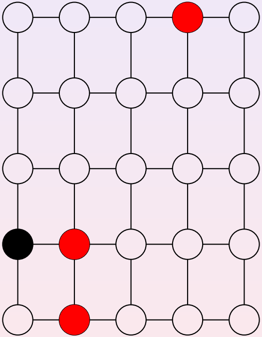
Initialization:

1. $\mathcal{C}$ places the cops;
2. $\mathcal{R}$ places the robber.

Step-by-step:

- each cop traverses at most 1 edge;
- the robber traverses at most 1 edge.

Robber captured:
A cop occupies the same vertex as the robber.



Cops & robber games

[Nowakowski and Winkler; Quilliot, 83]

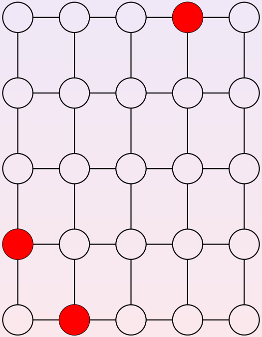
Initialization:

1. $\mathcal{C}$ places the cops;
2. $\mathcal{R}$ places the robber.

Step-by-step:

- each cop traverses at most 1 edge;
- the robber traverses at most 1 edge.

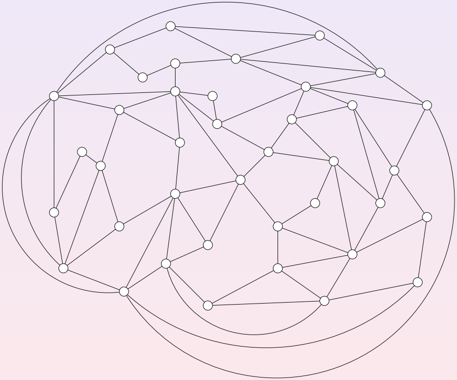
Robber captured:
A cop occupies the same vertex as the robber.



Cop number

$cn(G)$ minimum number of cops to capture any robber

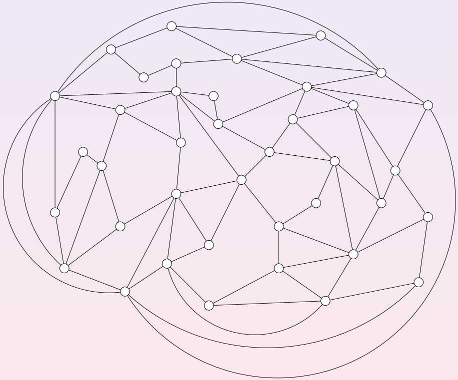
Determine $cn(G)$ for the following graph G?



Cop number

$cn(G)$ minimum number of cops to capture any robber

Determine $cn(G)$ for the following graph G? ≤ 3



$cn(G) \leq 3$ for any planar graph G [Aigner, Fromme, 84]

Computing $cn(G)$ is NP-hard [Fomin,Golovach,Kratochvil,N.Suchan'10]

Cops & robber games vs. graph structure

- G with **girth** g (min induced cycle) and **min degree** d: $cn(G) \geq d^g$ [Frankl 87]
- $\exists$ n-node graphs G (projective plane): $cn(G) = \Theta(\sqrt{n})$ [Frankl 87]
- G with **dominating set** k: $cn(G) \leq k$ [folklore]
- **Planar graph** G: $cn(G) \leq 3$ [Aigner, Fromme, 84]
- **Minor free graph** G excluding a minor H: $cn(G) \leq |E(H)|$ [Andreae, 86]
- G with **genus** g: $cn(G) \leq 3/2g + 3$ [Schröder, 01]
- G with **treewidth** t: $cn(G) \leq t/2 + 1$ [Joret, Kaminsk,Theis 09]
- G with **chordality** k: $cn(G) \leq k - 1$ [Kosowski, Li, N. Suchan'12]
- G **random graph** (Erdös Reyni): $cn(G) = O(\sqrt{n})$ [Bollobas *et al.* 08]
- **any** n-node graph G: $cn(G) = O(\frac{n}{2^{\sqrt{\log n}}})$ [Lu,Peng 09, Scott,Sudakov 10]

Conjecture: For any connected n-node graph G, $cn(G) = O(\sqrt{n})$. [Meyniel 87]

Link with *hyperbolicity* (cf. David's talk)

Variant of cop-number provides an approximation of hyperbolicity [Chalopin *et al.*'13].

Since 25 years, many researchers **study graphs structural properties** and introduce variants in the game to try solving the conjecture (e.g., fast robber [Fomin,Golovach,Kratochvil,N.Suchan'10]).
e.g., [Chiniforooshan 08, Bonato *et al.* 10, FGKNS 10, Alon,Mehrabian11, CCNV11, Clarke,McGillivray11]
see the recent survey book: The Game of Cops and Robbers on Graphs, A.Bonato and R.Nowakovski 2011

Outline

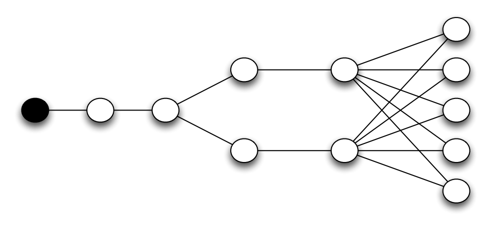
1. Graph Searching games
2. Cops and Robber games
3. Surveillance games

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded



Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

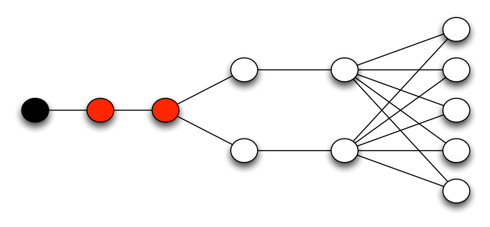
1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded



Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

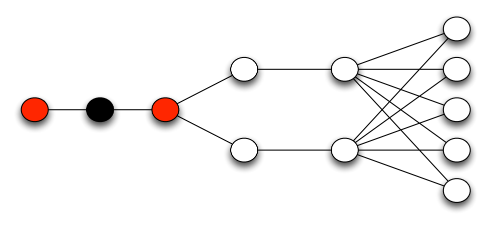
1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded



Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

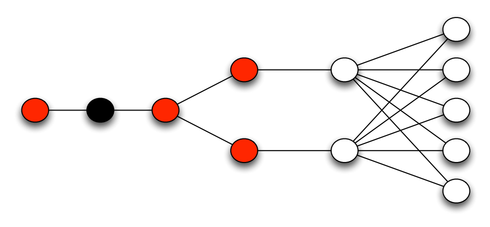
1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded



Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

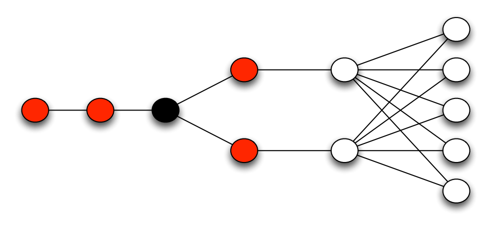
1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded



Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

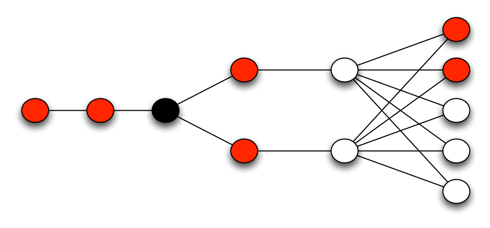
1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded



Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

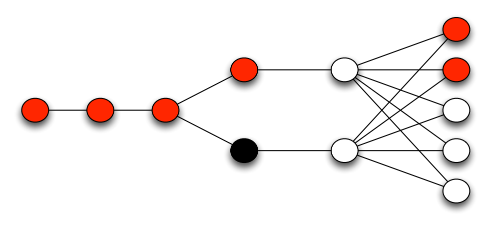
1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded



Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

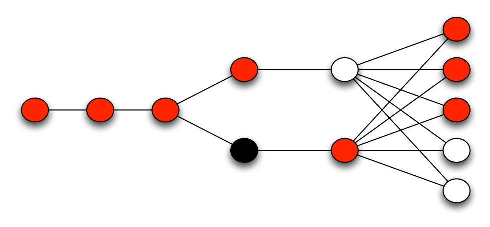
1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded



Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

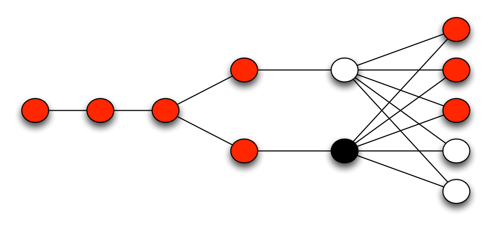
1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded



Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

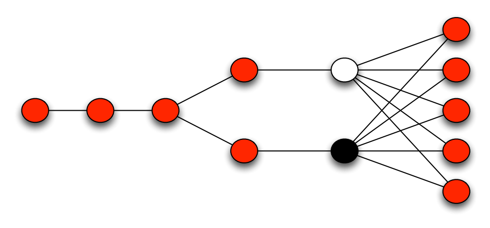
1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

Webpage Prefetching and Surveillance game

Model for Prefetching/Caching [Fomin *et al.*'12]

Web-surfer following hyperlinks. Webpage MUST be download before it arrives on it

bandwidth limitation: number of download per step is bounded

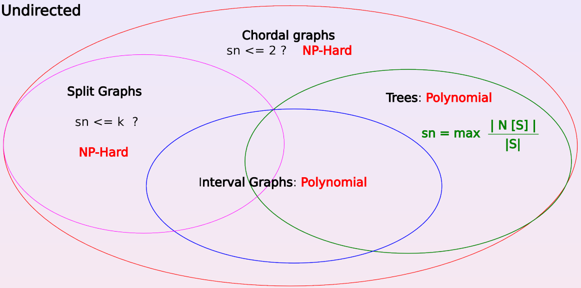


Initially, Fugitive (Websurfer) at some node, and **Turn-by-turn**

1. Web-browser prefetches $\leq k$ pages, i.e., marks $\leq k$ nodes
2. Fugitive may move on adjacent node

surveillance number(G, v_0) = min. number k of marks per turn avoiding Fugitive (starting from v_0) to reach an unmarked node (in the example = 2)

Surveillance game: results and open problems



Directed

DAGs: sn <=4 ? P-SPACE-Complete

DAGs: sn <=2 ? NP-Hard

General

(out)-degree (v0) ≤ sn ≤ max(degree (v0), max degree -1)

max out-degree

$O(2^n)$ exact algorithm

Online version: best strategy uses $\Theta(\Delta)$ marks per turn [Giroire *et al.*'13]

Connected version: set of marked nodes must always be connected

What is the cost of connectedness?

Conclusion

Many other "similar" games:

eternal vertex set, eternal domination, locating game, Lion and man, etc.

Other applications that could take advantage of these approach?

Most of these game are hard:

Computing optimal strategies are NP-hard (Graph Searching), PSPACE-complete (Surveillance Game) or even EXPTIME-complete (Cops and Robber), etc.
Few or no approximation algorithms are known!!

on-going work: Unified and generalized framework: Fractional games

- Algorithm to compute strategy using LP (winning states are polytopes)
- ...but some step of it is exponential (projection on subspace)

Complexity of computing fractional strategies?

A few very hard questions:

Meyniel conjecture: $O(\sqrt{n})$ cops are sufficient to capture a robber in n-node graphs?

Planar treewidth: Complexity of computing the treewidth in planar graphs?

Conclusion

Many other "similar" games:

eternal vertex set, eternal domination, locating game, Lion and man, etc.

Other applications that could take advantage of these approach?

Most of these game are hard:

Computing optimal strategies are NP-hard (Graph Searching), PSPACE-complete (Surveillance Game) or even EXPTIME-complete (Cops and Robber), etc.
Few or no approximation algorithms are known!!

on-going work: Unified and generalized framework: Fractional games

- Algorithm to compute strategy using LP (winning states are polytopes)
- ...but some step of it is exponential (projection on subspace)

Complexity of computing fractional strategies?

A few very hard questions:

Meyniel conjecture: $O(\sqrt{n})$ cops are sufficient to capture a robber in n-node graphs?

Planar treewidth: Complexity of computing the treewidth in planar graphs?

Conclusion

Many other "similar" games:

eternal vertex set, eternal domination, locating game, Lion and man, etc.

Other applications that could take advantage of these approach?

Most of these game are hard:

Computing optimal strategies are NP-hard (Graph Searching), PSPACE-complete (Surveillance Game) or even EXPTIME-complete (Cops and Robber), etc.
Few or no approximation algorithms are known!!

on-going work: Unified and generalized framework: Fractional games

- Algorithm to compute strategy using LP (winning states are polytopes)
- ...but some step of it is exponential (projection on subspace)

Complexity of computing fractional strategies?

A few very hard questions:

Meyniel conjecture: $O(\sqrt{n})$ cops are sufficient to capture a robber in n-node graphs?

Planar treewidth: Complexity of computing the treewidth in planar graphs?

Gracias !