Detecting an induced subdivision of K_4

Ngoc Khang Le

Team MC2, LIP, ENS de Lyon, Lyon, France

September 13, 2017

Outline

Outline

1. Introduction

2. Detecting ISK4

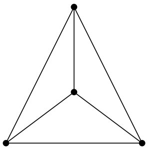
ISK4-free graphs

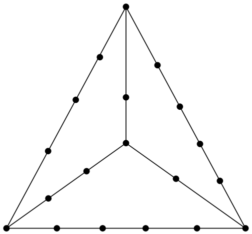
Definition

A *subdivision* of a graph is obtained by subdividing its edges into paths of arbitrary length (at least 1).

Definition

A graph is *ISK4-free* if it contains no induced subdivision of K_4.





ISK4-free graphs

Theorem (Le 2016)

Let G be an ISK4-free graph. Then $\chi(G) \leq 24$.

ISK4-free graphs

Theorem (Le 2016)

Let G be an ISK4-free graph. Then $\chi(G) \leq 24$.

Conjecture (Lévêque, Maffray, Trotignon 2012)

Let G be an ISK4-free graph. Then $\chi(G) \leq 4$.

ISK4-free graphs

Theorem (Le 2016)

Let G be an ISK4-free graph. Then $\chi(G) \leq 24$.

Conjecture (Lévêque, Maffray, Trotignon 2012)

Let G be an ISK4-free graph. Then $\chi(G) \leq 4$.

- Finding maximum clique is easy.
- Coloring ISK4-free graphs is NP-hard since it contains all line graphs of cubic graphs.
- Finding maximum stable set is open.

Detection problem

Problem 1

Input: A fixed graph H and a graph G.
Output: Decide if G contains some subdivision of H as a subgraph.

Detection problem

Problem 1

Input: A fixed graph H and a graph G.
Output: Decide if G contains some subdivision of H as a subgraph.

This is polynomial-time solvable by a theorem of Robertson and Seymour.

Detection problem

Problem 1

Input: A fixed graph H and a graph G.
Output: Decide if G contains some subdivision of H as a subgraph.

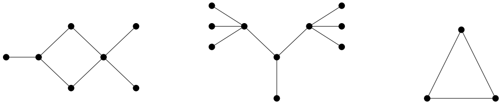
This is polynomial-time solvable by a theorem of Robertson and Seymour.

Problem 2

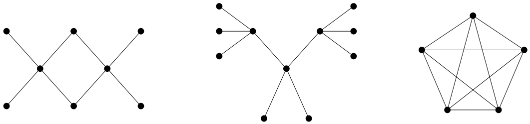
Input: A fixed graph H and a graph G.
Output: Decide if G contains some subdivision of H as an induced subgraph.

Detection problem

- Polynomial: (Chudnovsky, Seymour 2010)



- NP-hard: (Lévêque, Lin, Maffray, Trotignon 2009)

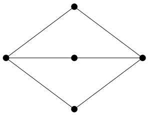


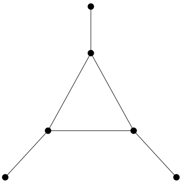
Problem

Open question

Do we have polynomial algorithm if H is a subcubic graph?

- True for following graphs: (Chudnovsky, Seymour, Trotignon 2013)





Our result

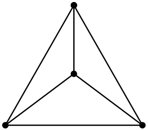
Theorem (Le 2017)

There is an $O(n^9)$ algorithm to detect an ISK4.

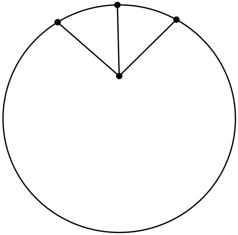
Outline

Detecting claw-free ISK4

- Detecting K_4 can be done in $O(n^4)$.
- Detecting twin-wheel can be done in $O(n^6)$.

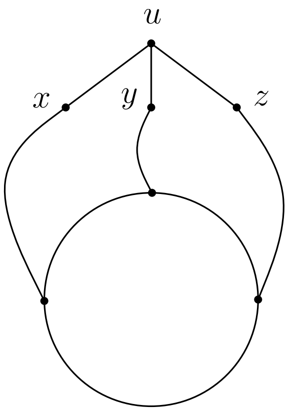


K_4

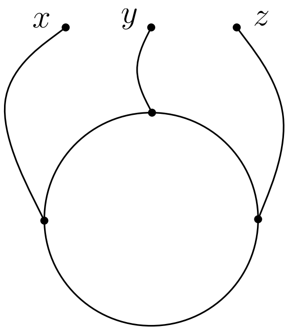


twin wheel

Detecting radar



Detecting radar



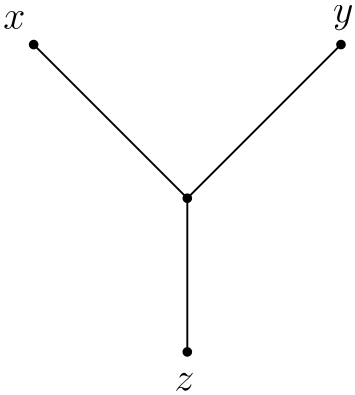
(x, y, z)-radar

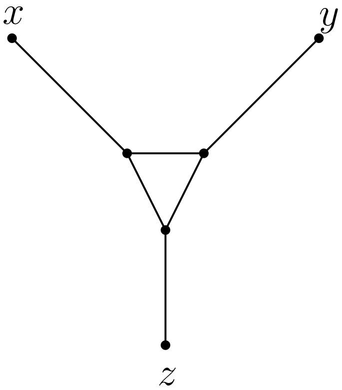
Our goal

Detecting (x, y, z)-radar in $G' = G \setminus (N[u] \setminus \{x, y, z\})$.

Minimum connected subgraph

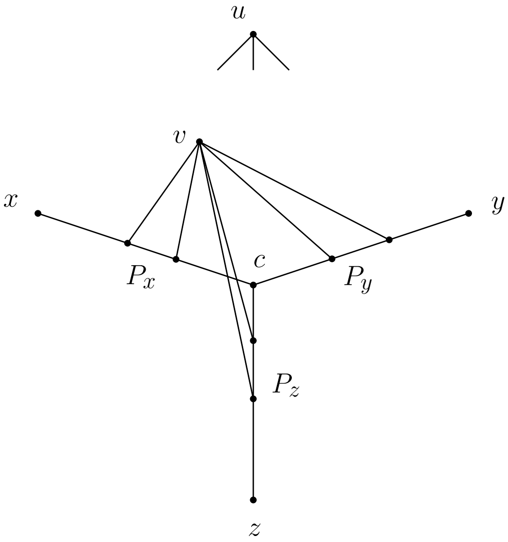
- Finding a minimum subgraph of G' connecting x, y, z in $O(n^3)$.





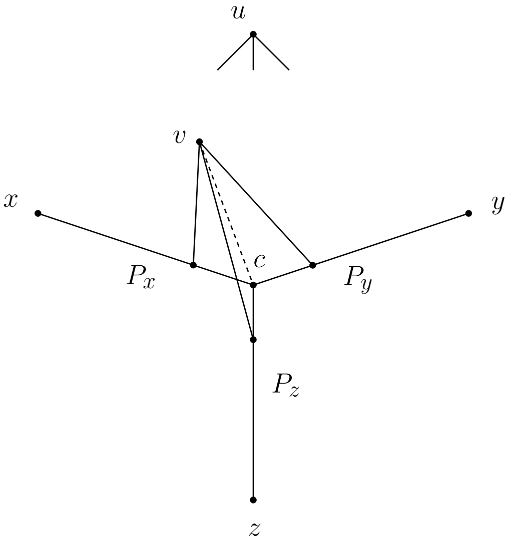
Attachment

v cannot have neighbors in all 3 paths.



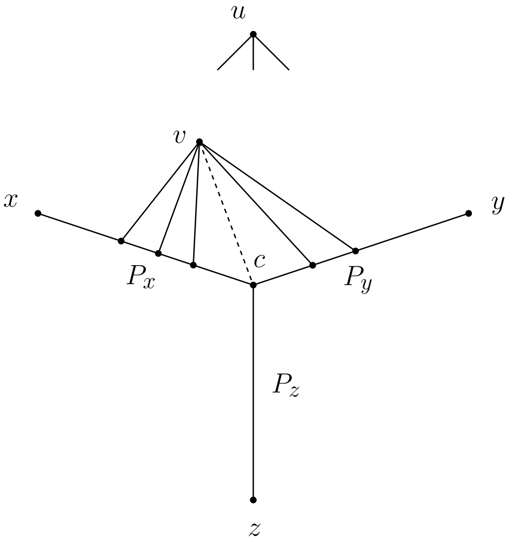
Attachment

v cannot have neighbors in all 3 paths.



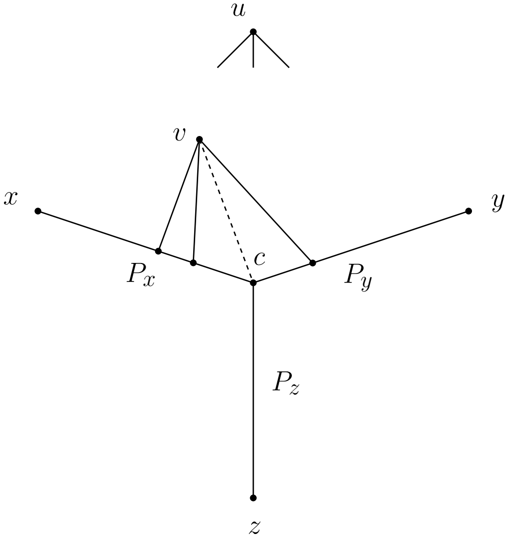
Attachment

v cannot have neighbors in some 2 paths.



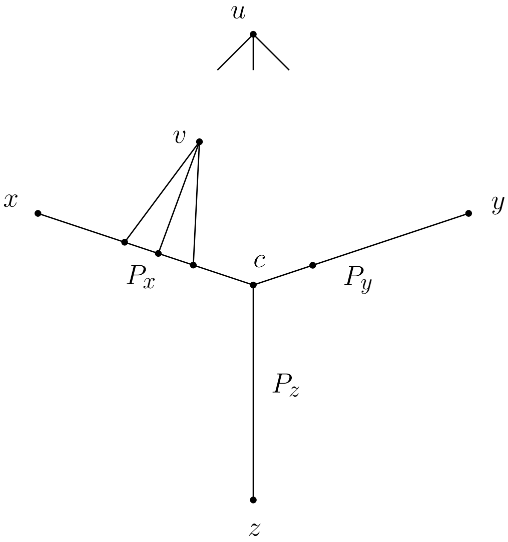
Attachment

v cannot have neighbors in some 2 paths.



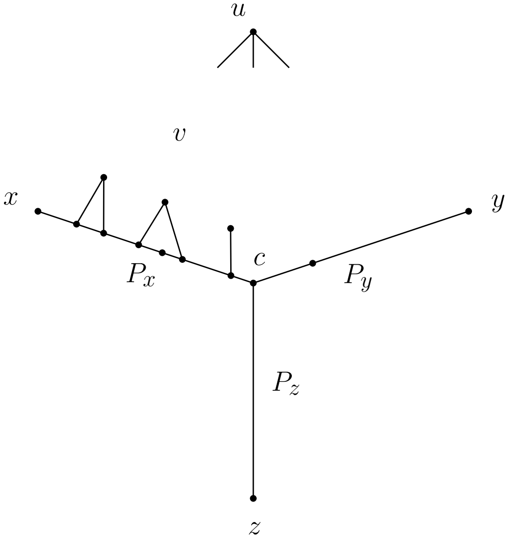
Attachment

v can only attach "locally" on each path.



Attachment

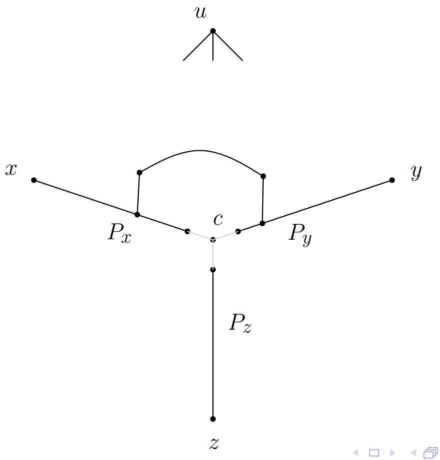
v can only attach "locally" on each path.



Short connection

Claim

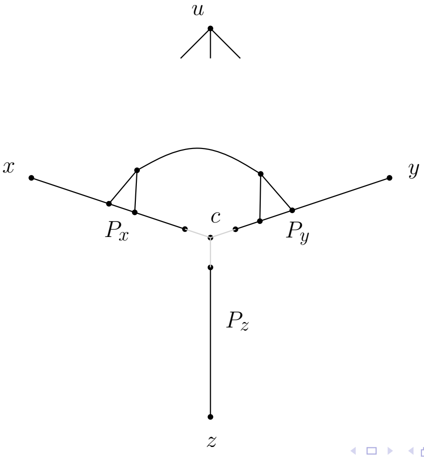
Every path from x to y in $G' \setminus c$ which contains at most 2 neighbors of c, certifies an ISK4 in G.



Short connection

Claim

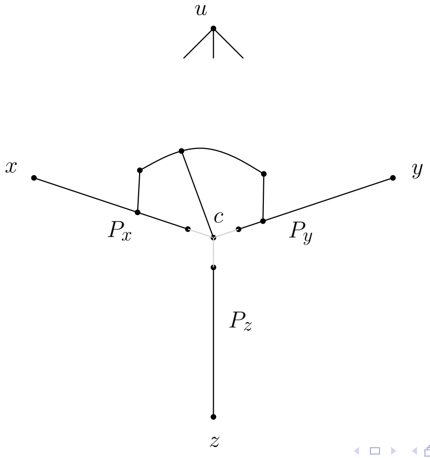
Every path from x to y in $G' \setminus c$ which contains at most 2 neighbors of c, certifies an ISK4 in G.



Short connection

Claim

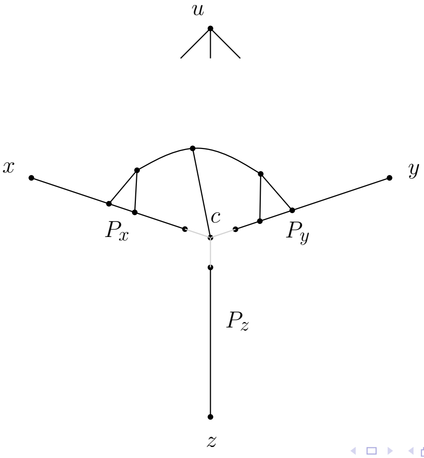
Every path from x to y in $G' \setminus c$ which contains at most 2 neighbors of c, certifies an ISK4 in G.



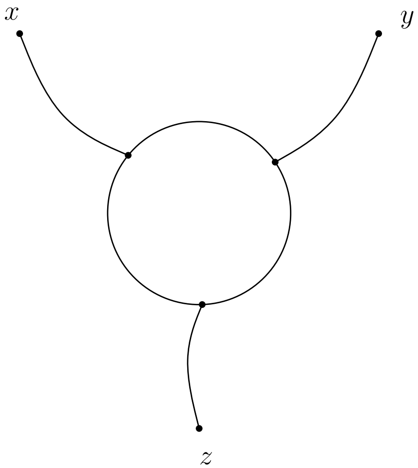
Short connection

Claim

Every path from x to y in $G' \setminus c$ which contains at most 2 neighbors of c, certifies an ISK4 in G.



c has at least 3 neighbors in every path among x, y, z.
$\Rightarrow$ c is not contained in any (x, y, z)-radar.



Sum up

(1) Check if G contains K_4 or twin-wheel.

(2) For every claw (u, x, y, z) in G:

  (a) Find a minimum subgraph H of $G' = G \setminus (N[u] \setminus \{x, y, z\})$ connecting x, y, z. Let c be the center of H.

  (b) Check if there exists some path in $G' \setminus c$ between some pair of $\{x, y, z\}$ which contains at most 2 neighbors of c.

  (c) $G := G \setminus c$, go to (a).

Complexity: $O(n^9)$.

Improvement

The complexity can be improved to $O(n^7)$ by:

- First decompose the graph by clique cutset until there is no $K_{3,3}$.
- (ISK4, $K_{3,3}$)-free has linear number of edges, so testing the connection takes only $O(n)$.
- Consider only $O(n^3)$ triples of independent vertices instead of generating all $O(n^4)$ claws.

Thank you !