Extensions of Subgraph Reconfiguration

Jeffrey Kam

University of Waterloo

May 2022

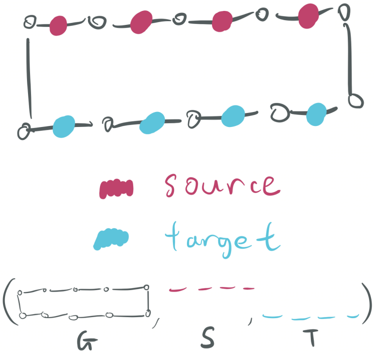
Introduction

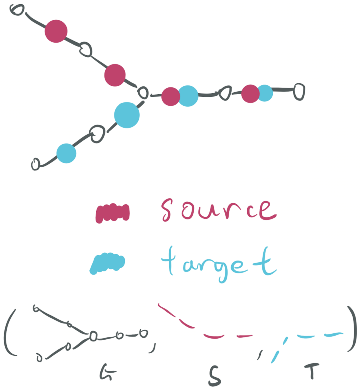
What is subgraph reconfiguration [Hanaka et al., 2020]?

- Given a graph G, a graph property Π, we want to reconfigure among subgraphs of G with property Π.
- 3 Rules: Token Sliding [Hearn and Demaine, 2005], Token Jumping [Kamiński et al., 2012], and Token Addition/Removal [Ito et al., 2011].
- 2 Variants: Edge-variant, and Vertex-variant [Hanaka et al., 2020].

Example under Edge-variant and TJ Rule

The property Π: "The graph is a path".





The talk will be restricted to the edge-variant and TJ rule.

Main Results

Property	Edge Variant - TJ
Path	NP-hard [Hanaka et al., 2020]
Cycle	P [Hanaka et al., 2020]
Tree	P [Hanaka et al., 2020]
k-bounded Path-width Tree	NP-hard [Theorem 1]

k-Bounded Path-width Tree Reconfiguration

Definition

A graph G is a *k-bounded path-width tree* if G is a tree such that its pathwidth is k or less.

Definition

Given a graph G and $v \in V(G)$, the HAMILTONIAN v-PATH problem asks whether G has a spanning path starting at v.

Theorem

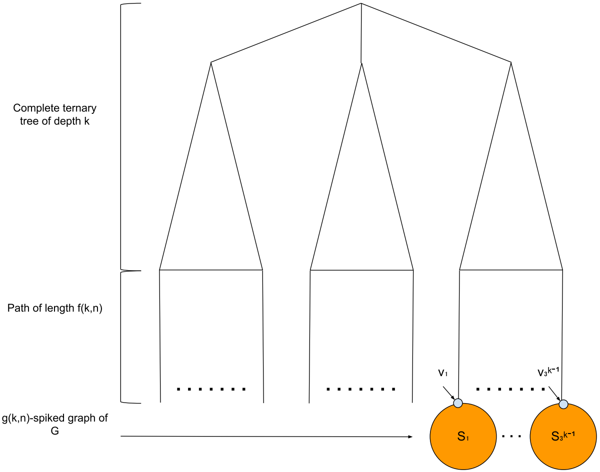
The k-BOUNDED PATH-WIDTH TREE RECONFIGURATION problem is NP-hard for any fixed $k \geq 1$.

Proof Overview:

1. Given (G, v) for the HAMILTONIAN v-PATH problem, construct an instance (G', S, T) for the reconfiguration problem.

Constructing (G', S, T): Overview

Given (G, v). Let $n = |V(G)|$.

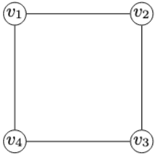


where $g(k, n) = 3^{k-1}n^3$, $f(k, n) = 3^{k-1} \cdot (n - 1 + n \cdot g(k, n)) + 2$.

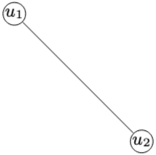
Constructing (G', S, T): (k, l)-claw

Definition

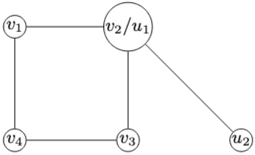
1-sum is an operation that joins 2 graphs by identifying a vertex from each graph and merging them into a single vertex.



(a) G_1



(b) G_2



(c) $G_1 \oplus_{v_2,u_1} G_2$

Definition

A (k, l)-*claw* is a ternary tree of depth k where each leaf node is 1-summed with an endpoint of a path of length $l - 1$ (called a tail).

Constructing (G', S, T): (k, l)-claw

Definition

An *almost (k, l)-claw* is a (k, l)-claw with a tail removed.

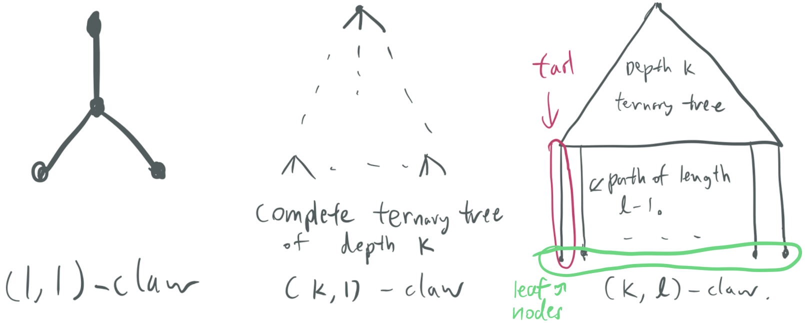


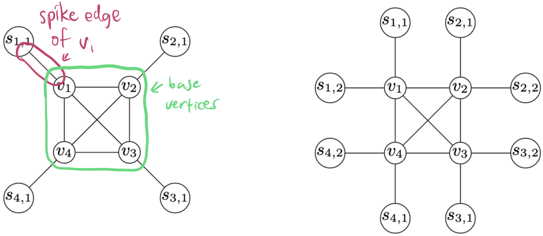
Figure: Example of (k, l)-claws and almost (k, l)-claws

The endpoints of the tails are called the *leaf nodes* of the (k, l)-claw.

Constructing (G', S, T): Spiked Graph

Definition

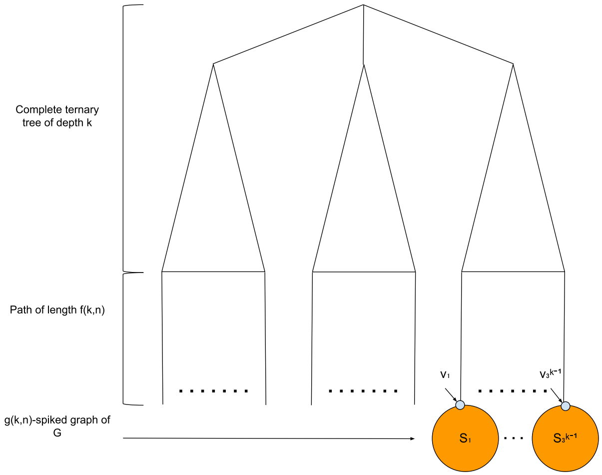
Let $G = (V, E)$. We define the *k-spiked graph of G* by adding k new pendant nodes as neighbours of each vertex in the graph.



(a) 1-spiked graph of K_4 (b) 2-spiked graph of K_4

We call the vertices of G the *base vertices*. For a base vertex v, the added edges incident to v are called the *spike edges* of v.

Constructing (G', S, T): G' revisited



where $g(k, n) = 3^{k-1} n^3$, $f(k, n) = 3^{k-1} \cdot (n - 1 + n \cdot g(k, n)) + 2$.

Constructing (G', S, T): Source and Target Tokens

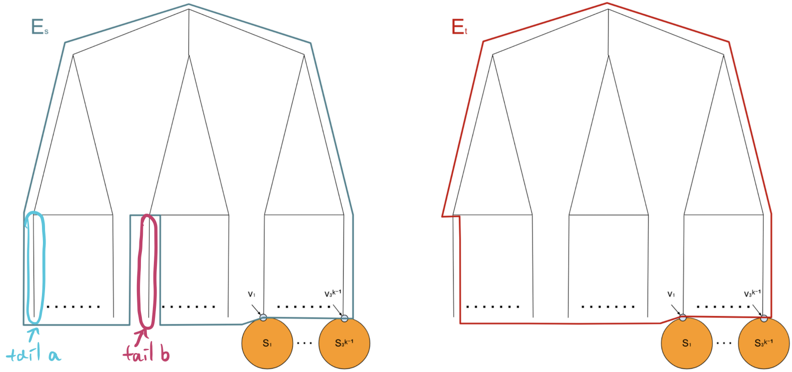


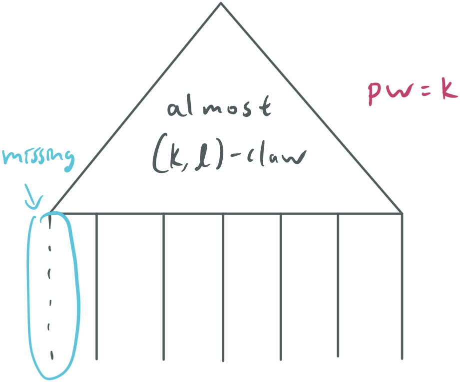
Figure: Source subgraph E_s (left) and Target subgraph E_t (right).

Constructing (G', S, T): Validity of Source and Target

Both source and target induce an almost (k, l)-claw.

Lemma

For $k \geq 1$ and $l \geq 2$, an almost (k, l)-claw has path-width k.



Constructing (G', S, T): Validity of Source and Target

Both source and target induce an almost (k, l)-claw.

Lemma

For $k \geq 1$ and $l \geq 2$, an almost (k, l)-claw has path-width k.

Definition

Let $k \geq 1$ and T be a tree. A vertex $v \in V(T)$ is k-good if $T \setminus v$ results in at most 2 branches with pathwidth k or more.

Theorem ([Scheffler, 1990])

Let $k \geq 1$ and T be a tree. $pw(T) \leq k$ if and only if v is k-good for all $v \in V(T)$.

k-Bounded Path-width Tree Reconfiguration

Theorem

The k-BOUNDED PATH-WIDTH TREE RECONFIGURATION problem is NP-hard for $k \geq 1$.

Proof Overview:

1. Given (G, v) for the HAMILTONIAN v-PATH problem, construct an instance (G', S, T) for the reconfiguration problem.
2. Show that G has a Hamiltonian v-path if and only if (G', S, T) is reconfigurable.

($\Rightarrow$) G has a Hamiltonian v-path

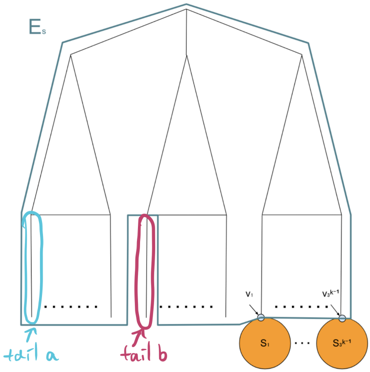


Figure: Source subgraph E_s (left) and Target subgraph E_t (right).

$(\Rightarrow)$ G has a Hamiltonian v-path

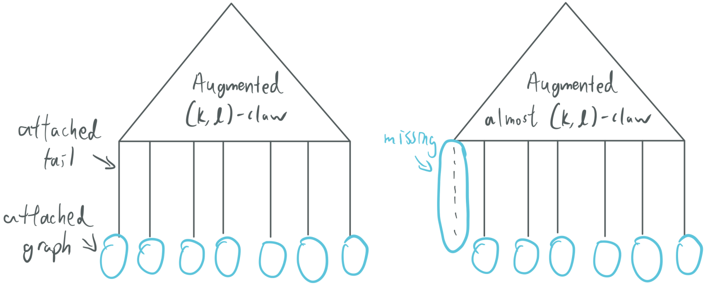
We show a reconfiguration sequence from S to T.

1. Move tokens from the tail a to fill the Hamiltonian v-paths in each of the spiked graphs.
2. Move the remaining tokens on tail a to the spike edges in each of the spiked graphs, until one token remains on tail a.

$(\Rightarrow)$ G has a Hamiltonian v-path

Definition

An augmented (k, l)-claw is a (k, l)-claw where each leaf node is 1-summed with a graph.



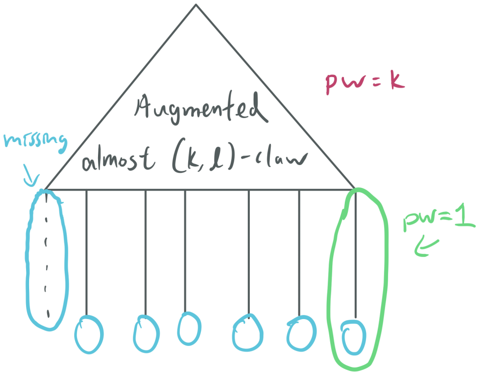
Definition

An augmented almost (k, l)-claw is an augmented (k, l)-claw where an attached graph and its attached tail are removed.

($\Rightarrow$) G has a Hamiltonian v-path

Lemma

Let $k \geq 1$ and $l \geq 2$. Let T be an augmented almost (k, l)-claw. If each attached graph together with its attached tail is of pathwidth 1, then T has pathwidth k.



$(\Rightarrow)$ G has a Hamiltonian v-path

We show a reconfiguration sequence from S to T.

1. Move tokens from the tail a to to fill the Hamiltonian v-paths in each of the spiked graphs.
2. Move the remaining tokens on tail a to the spike edges in each of the spiked graphs, until one token remains on tail a.
3. Move the last token on tail a to tail b.
4. Revert steps 2 and 1 but move the tokens to tail b this time.

$(\Leftarrow)$ (G', S, T) is Reconfigurable

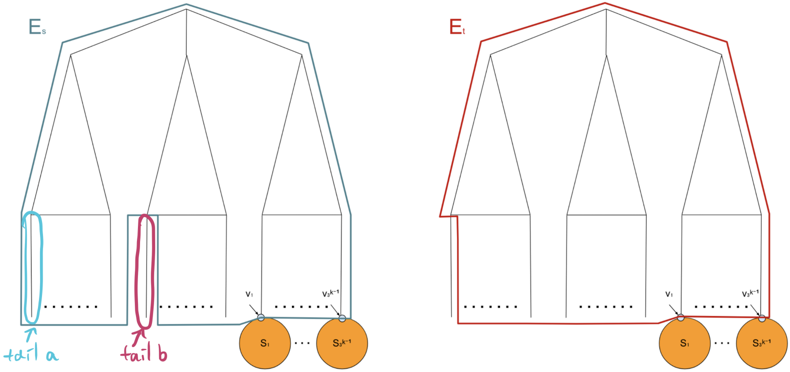


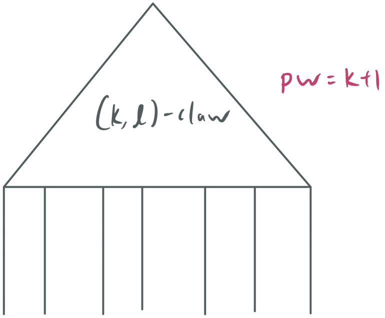
Figure: Source subgraph E_s (left) and Target subgraph E_t (right).

$(\Leftarrow)$ (G', S, T) is Reconfigurable: Step 1

- When we place tokens on tail b, there is at least 1 other tail where no tokens are placed on it.

Lemma

For $k \geq 1$ and $l \geq 2$, a (k, l)-claw has path-width $k + 1$.

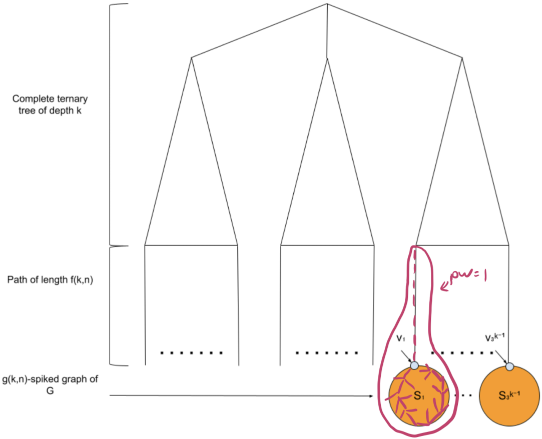


$(\Leftarrow)$ (G', S, T) is Reconfigurable: Step 1

- When we place tokens on tail b, there is at least 1 other tail where no tokens are placed on it.
- At least $f(k, n) - 1$ tokens have to move from the source to edges of the spiked graph.
- By a counting argument, for each base vertex v, there is a token on at least one spike edge of v.

$(\Leftarrow)$ (G', S, T) is Reconfigurable: Step 2

- There is a spiked graph and its attached tail where the tokens placed on them induce a subgraph of pathwidth 1.



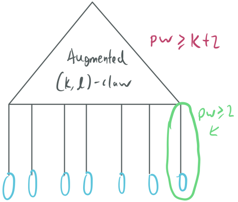
($\Leftarrow$) (G', S, T) is Reconfigurable: Step 2

Lemma

Let $k \geq 1$ and $l \geq 2$. Let T be an augmented (k, l)-claw. If each attached graph H satisfies

- *H is a tree, and*
- *H together with its attached tail is of pathwidth 2 or more,*

then T has pathwidth $k + 2$ or more.



$(\Leftarrow)$ (G', S, T) is Reconfigurable: Step 3

- By the spiked graph lemma, there is a Hamiltonian v-path in G.

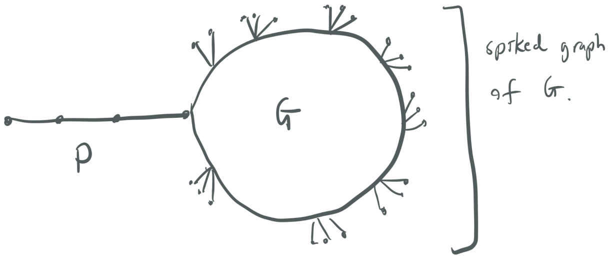
Lemma (Spiked Graph Lemma)

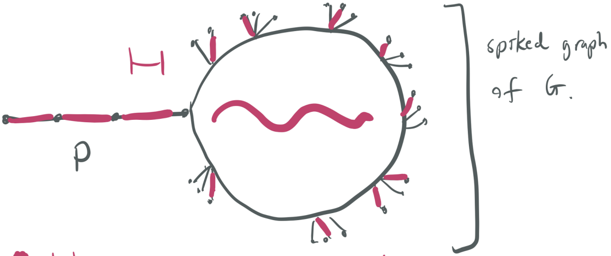
Let G be a graph. Let S be the k-spiked graph of a graph G. Let P be a path of length 2 or more with an endpoint p. Let G' be a graph by 1-summing p and some base vertex of S. If H is a subgraph of G' satisfying

1. *H is a connected,*
2. *H is of pathwidth 1,*
3. *$E(H)$ contains all edges of P, and*
4. *$E(H)$ contains at least a spike from each vertex of G,*

then G contains a Hamiltonian path starting at v.

Proof Sketch of Spiked Graph Lemma





H is a connected subgraph of pathwidth 1

Future Direction

- Allowing moving k tokens simultaneously for some fixed k.

Π is minor-closed. (G, S, T) an instance of Π-RECONFIGURATION.

- Define the quantity "minimum running buffer" (MRB) [Gao et al., 2021] of (G, S, T) as the minimum of the maximum number of token that needs to be placed in a buffer in order to reconfigure.
- MRB of Π-RECONFIGURATION = maximum MRB over all possible instances where G is connected.
  - For FOREST-RECONFIGURATION, $MRB = 0$.
  - For CACTUS-RECONFIGURATION, we think $MRB = 1$.
  - For PLANAR-RECONFIGURATION, we think $MRB = O(|S|)$.
- Hope to classify graph properties that have bounded MRB.

References I

[Gao et al., 2021] Gao, K., Feng, S. W., and Yu, J. (2021).
On minimizing the number of running buffers for tabletop rearrangement.
ArXiv, abs/2105.06357.

[Hanaka et al., 2020] Hanaka, T., Ito, T., Mizuta, H., Moore, B., Nishimura, N., Subramanya, V., Suzuki, A., and Vaidyanathan, K. (2020).
Reconfiguring spanning and induced subgraphs.
Theoretical Computer Science, 806:553–566.

[Hearn and Demaine, 2005] Hearn, R. A. and Demaine, E. D. (2005).
Pspace-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation.
Theor. Comput. Sci., 343:72–96.

References II

[Ito et al., 2011] Ito, T., Demaine, E. D., Harvey, N. J., Papadimitriou, C. H., Sideri, M., Uehara, R., and Uno, Y. (2011).
On the complexity of reconfiguration problems.
Theoretical Computer Science, 412(12-14):1054–1065.

[Kamiński et al., 2012] Kamiński, M., Medvedev, P., and Milanič, M. (2012).
Complexity of independent set reconfigurability problems.
Theoretical Computer Science, 439:9–15.

[Scheffler, 1990] Scheffler, P. (1990).
A linear algorithm for the pathwidth of trees.
In *Topics in combinatorics and graph theory*, pages 613–620. Springer.