Reconfiguring Connected Subgraphs with Path-width $\leq k$

Jeffrey Kam [i] Steven Lok Ngai Wong [ii]

December 17, 2021

Abstract

Within the family of Subgraph reconfiguration problems, we prove a new complexity result for caterpillar subgraphs, and attempt to prove the same result for a closely related graph property. Specifically, we show that the edge variant of the Caterpillar reconfiguration problem is NP-hard under the TJ rule, and attempt to prove the same result for the k-bounded path-width tree reconfiguration problem.

1 Introduction

Consider a graph where each node corresponds to a feasible solution of an instance of a search problem P, and edges exist between two nodes if the feasible solutions corresponding to the nodes are "adjacent" according to some *reconfiguration rule* $\mathcal{A}$. We call this the *reconfiguration graph* for P and $\mathcal{A}$. In the *reachability problem* for P and $\mathcal{A}$, given *source* and *target* solutions to P, we want to determine whether or not there exists a path between their corresponding nodes. If such a path exists, we call this the *reconfiguration sequence* between the source and target solutions, where each edge on the path corresponds to a *reconfiguration step*.

In particular, subgraph reconfiguration describes a family of reachability problems where feasible solutions are subgraphs (of an input graph) that satisfy a specified graph structure property Π. Each problem in the family can be specified by how the the node set and edge set are defined in the reconfiguration graph. Note that we use the term *node* for reconfiguration graphs and *vertex* for input graphs.

If a feasible solution (subgraph and hence node) is represented by an edge subset of the input graph, we call this the *edge variant*. There is also the *induced variant* and *spanning variant* for when a subgraph is represented by a vertex subset, which we omit for brevity.

Since a feasible solution is represented by a subset of edges or nodes, we can consider that there are tokens placed on the edges or nodes in the subset. A reconfiguration step (edge) can then be described by rules for how these tokens can be moved or changed. One rule is called *token-jumping* (TJ), where a token can move to any other unoccupied edge or node. Another rule is called *token-sliding* (TS), where a token can move to any other adjacent edge or node (we say two edges are adjacent if they share a common vertex). Lastly, there is also the *token-addition-and-removal* rule (TAR), where in one step we can either add or remove a token [HIM+20].

[i] University of Waterloo, email: hykam@uwaterloo.ca.

[ii] University of Waterloo, email: slnwong@uwaterloo.ca.

In this paper, we study the complexity of subgraph reconfiguration for the edge variant under the TJ rule for subgraphs with the property of being connected and with path-width $\leq k$ for some fixed k. We first prove a result for when the subgraph is of path-width 1, i.e. a caterpillar. Then, we attempt to generalize our results to trees with path-width $\leq k$.

1.1 Main Result and Proof Overview

Our main result is Theorem 3.4 − that the Caterpillar reconfiguration problem under the TJ rule is NP-hard. An overview of the proof is that we want to reduce the Hamiltonian v-path problem to the Caterpillar reconfiguration problem in polynomial time. For this reduction:

1. We create an auxiliary graph G' of polynomial size from the input graph G for which we want to solve the Hamiltonian v-path problem.
2. We show that G has a Hamiltonian path starting at v if and only if there exists a reconfiguration sequence between a source and target subgraph of our choosing in G'.

1.2 Related Work

The Subgraph reconfiguration problem was first proposed by Hanaka et al. [HIM+20], which includes results for graph properties such as paths, trees, and cycles. There has also been some work on a related path reconfiguration problem under the TS rule by Demaine et al. [DEH+19], but their sliding rule concerns sliding the path as a whole, as opposed to sliding individual tokens. Here we show a summary of our contributions and the current results related the problem.

Property	Edge Variant - TJ	Edge Variant - TS
Path	NP-hard [HIM+20]	Unknown
Cycle	P [HIM+20]	P [HIM+20]
Tree	P [HIM+20]	Unknown
Caterpillar	NP-hard [Theorem 3.4]	Unknown
k-bounded Path-width Tree	NP-hard? [Theorem 3.15]	Unknown
Any Property	XP [HIM+20]	XP [HIM+20]

Note that for ?, we have not yet completed the entire proof as some sections only have proof sketches.

1.3 Organization

In Section 3.1, we provide a proof of the NP-hardness of the Caterpillar reconfiguration problem. In Section 3.2, we provide a proof of the NP-hardness of the k-bounded path-width tree reconfiguration problem. In Section 4, we conclude and present some open problems.

2 Preliminaries

2.1 General Facts and Notations

Let Π be a graph property (e.g. "a graph is a path"). We will refer to the edge variant of Subgraph reconfguration for the property Π as the Π reconfiguration problem. Let G be an input graph which is simple. Then, a tuple (G, E_s, E_t) is an *instance of the* Π RECONFIGURATION *problem*, where E_s, E_t are the edge subsets for the source and target solutions respectively, when the subgraph induced by E_s and E_t satisfies property Π. We say that (G, E_s, E_t) is *reconfigurable* if there exists a reconfiguration sequence from E_s to E_t. Otherwise, we say it is *not reconfigurable*.

A *Hamiltonian path* in a graph G is a path that visits every vertex in G exactly once. The HAMILTONIAN v-PATH problem is a decision problem for whether or not a Hamiltonian path starting at the vertex v exists in a given graph G. The HAMILTONIAN v-PATH problem is NP-complete by a simple reduction to the standard HAMILTONIAN PATH problem, and is still NP-complete when restricted to connected and non-empty graphs.

A *caterpillar* is a tree in which the removal of all the leaf vertices of the tree results in a path. We call the resulting path the *spine* of the caterpillar, and we call the removed vertices the *legs* of the caterpillar. Moreover, if a leg is adjacent to an endpoint of the spine, we call that leg an *end* of the caterpillar. One characterization of caterpillars are connected graphs of path-width 1.

Let $G = (V, E)$ be a graph, and fix an ordering of the vertices $v_1, \cdots, v_n$. Let $P = p_1 p_2 \cdots p_m$ be a path. For each $p_i \in V(P)$, define a *bag* $B_i \subseteq V$ to be a subset of the vertex set. The sequence $(B_i : 1 \leq i \leq m)$ *path decomposition* of G if the following holds

1. For $uv \in E$, $\{u, v\} \subseteq B_i$ for some $i \in [n]$, and
2. Let $u, v \in V$. If $u \in B_i$ and $u \in B_j$ for some $i, j \in [n]$ and $i \leq j$, then $u \in B_k$ for all $i \leq k \leq j$.

The *width* of the path decomposition is $\max_{v_i \in V}(|B_i| + 1)$. The *path-width of* G, denoted $pw(G)$, is defined as the minimum width of all path decomposition of G [RS83].

A *k-bounded path-width tree* G is a tree that is connected and that $pw(G) \leq k$.

Notation. Let G_1 and G_2 be distinct graphs with at least one vertex, and let $u \in G_1$, $v \in G_2$. Then the 1-*sum* $G_1 \oplus_{u,v} G_2$ is the graph obtained by merging vertex u with vertex v. We refer to this merged vertex as either u or v interchangeably.

Notation. Let G be a graph, $T \subseteq E(G)$, and $S \subseteq V(G)$. We define the following operations:

1. $G \setminus_E T$ denotes the removal of all edges of T in G and then removing any isolated vertices.
2. $G \setminus_V S$ denotes the induced subgraph of G given by vertex set $V(G) \setminus S$.

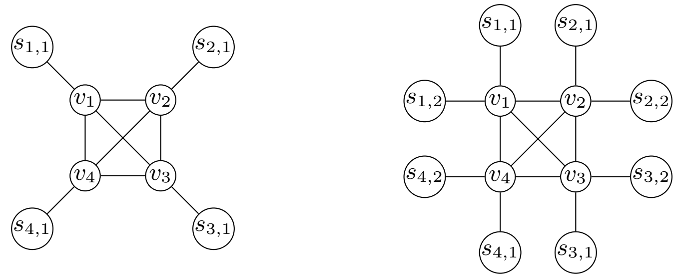
3 Technical Section

3.1 Caterpillar Reconfiguration

Definition 3.1. Let $G = (V, E)$ be a graph where $V = \{v_1, \cdots, v_n\}$. We define the *k-spiked graph of* G as S where

$$V(S) = V \cup \bigcup_{i=1}^{n} \{s_{i,j} : 1 \leq j \leq k\} \text{ and } E(S) = E \cup \bigcup_{i=1}^{n} \{v_i s_{i,j} : 1 \leq j \leq k\}.$$

This is the graph G with k new vertices and edges for each existing vertex. We call each $v_i s_{i,j}$ a *spike* of G.



(a) 1-spiked (b) 2-spiked

Figure 1: k-spiked graphs of K_4

Lemma 3.2. *Let S be the k-spiked graph of a graph G and let v be a particular vertex of G. Let P be a path of length ≥ 2, and let p_1, p_2 be the endpoint vertices of P. Consider the graph $P \oplus_{p_2,v} S$ and let H be a subgraph of this graph.*

Then if H is a caterpillar that contains all of P and uses at least one spike of each vertex in G, then G contains a Hamiltonian path starting at v.

Proof. Note that as a caterpillar is a tree, and H uses at least one spike of each vertex in G, we must have that H contains a spanning tree of G.

In H, considering following the path starting at p_1 until the path diverges (some vertex has degree ≥ 3), or until the path ends (some vertex has degree 1).

Suppose for a contradiction that the path diverges at a vertex t of degree ≥ 3. Note that the earliest vertex of divergence is v. As the path has length ≥ 2, let w be the vertex encountered in the path prior to t and note that w has degree ≥ 2. Let tu_1, tu_2 be two edges from the divergence of the path. Then as each vertex of G uses at least one spike, in particular u_1 and u_2 each use a spike and therefore have degree ≥ 2. Then this means that w, t, u_1, u_2 are all in the spine of H. But the spine is a path, and t still has degree ≥ 3, a contradiction.

Therefore, we must have that the path ends. So part of H forming a spanning tree of G actually forms a path using all the vertices of G, with the vertex v as one of it's endpoints. So G contains a Hamiltonian path starting at v. □

Definition 3.3. Let $G = (V, E)$ be a graph where $V = \{v_1, \cdots, v_n\}$. Let v be a particular vertex of G. We define the auxiliary graph $G'_{cater}(v)$ as follows:

- Let $P_1 = p^1_1 \cdots p^1_{2n-1}$ be a path of length $2n - 1$.
- Let $P_2 = p^2_1 \cdots p^2_{2n-1}$ be a path of length $2n - 1$.
- Let Y be a claw (a star with 3 edges) with vertices $\{a, b, c, r\}$, where r is the center vertex, and with an additional vertex d and edge cd.
- Let S be the 1-spiked graph of G.

Then $G'_{cater}(v)$ is the graph formed by connecting S, P_1, P_2 by adding the edges $p^1_{2n-1}a$, $p^2_{2n-1}b$, and dv.

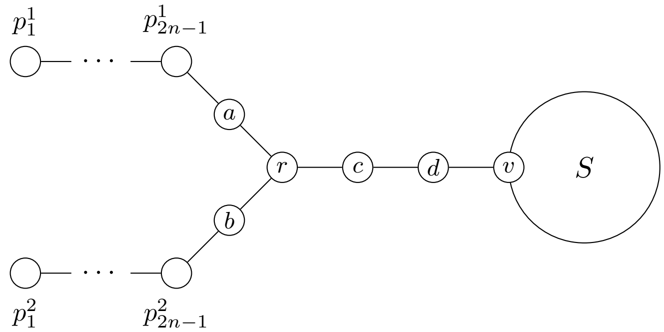


Figure 2: $G'_{cater}(v)$

Theorem 3.4. *The* Caterpillar reconfiguration *problem under the TJ rule is* NP*-hard.*

Proof. Since the Hamiltonian v-path problem is NP-hard, to show that the Caterpillar reconfiguration problem is NP-hard, it suffices to give a polynomial-time reduction from the Hamiltonian v-path problem.

Let (G, v) be an instance of the Hamiltonian v-path problem. Using G, we create the auxiliary graph $G'_{cater}(v)$ from 3.3. Let

$$E_s = E(P_1) \cup E(Y) \cup \{p_{2n-1}^1 a, dv\} \text{ and } E_t = E(P_2) \cup E(Y) \cup \{p_{2n-1}^2 b, dv.$$

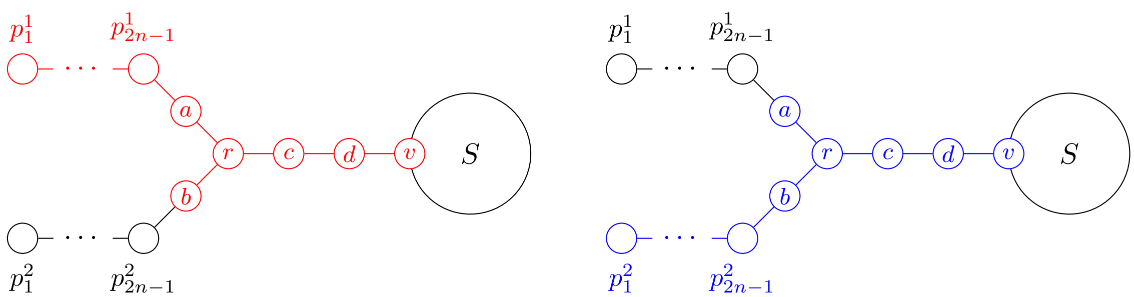


Figure 3: E_s and E_t in $G'_{cater}(v)$

Then we want to show that G has a Hamiltonian path starting at v if and only if the instance $(G'_{cater}(v), E_s, E_t)$ is reconfigurable.

($\Rightarrow$) Suppose that G has a Hamiltonian path starting at v. Let this Hamiltonian path be $e_1 \cdots e_{n-1}$, where e_1 is the edge starting at v.

Note that the number of edges on this path is $n-1$, and that the number of spikes in $G'_{cater}(v)$ is n. Then one possible reconfiguration sequence from E_s to E_t in $G'_{cater}(v)$ is:

(a) First, move (token jump) $p_1^1 p_2^1$ to e_1, $p_2^1 p_3^1$ to e_2, $\cdots$, and $p_{n-1}^1 p_n^1$ to e_{n-1}.

(b) Then, move $p_n^1 p_{n+1}^1$ to $v_1 s_1$, $p_{n+1}^1 p_{n+2}^1$ to $v_2 s_2$, $\cdots$, $p_{2n-2}^1 p_{2n-1}^1$ to $v_{n-1} s_{n-1}$, and $p_{2n-1}^1 a$ to $v_n s_n$.

Each node in the reconfiguration sequence so far satisfies the property Π because at each step, the removal of all the leaves in the subgraph formed by the tokens results in a path (we first move the spine, and then convert spine edges to legs).

Then, we can symmetrically reverse the reconfiguration steps from steps (a) and (b) but for the edges in P_2 (i.e. use p_i^2 instead of p_i^1), which maintains the Π property and results in the target solution.

Therefore, there exists a reconfiguration sequence from the source solution to the target solution, so $(G'_{cater}(v), E_s, E_t)$ is reconfigurable.

($\Leftarrow$) Suppose that $(G'_{cater}(v), E_s, E_t)$ is reconfigurable. This means that there exists a node in the reconfiguration sequence in which a token is moved to $p_{2n-1}^2 b$, as this edge is part of the target solution. Consider the placement of tokens at this node.

Note that until (at least) the edges $p_1^1 p_2^1, \cdots, p_{2n-2}^1 p_{2n-1}^1$, and $p_{2n-1}^1 a$ are moved, we cannot move any token to $p_{2n-1}^2 b$, as otherwise the subgraph induced by the edges would not satisfy Π.

This is a total of $\underbrace{(2n-2)}_{E(P_1)} + \underbrace{1}_{p_{2n-1}^1 a} = 2n-1$ tokens that need to have been moved. The only place for these tokens to move to are in S, as the subgraph at each step must be connected.

Also, note that a caterpillar does not have cycles, as a caterpillar is a tree. So the maximum number of tokens that can be placed in S while maintaining Π is $(n-1)+n = 2n-1$, which is when the tokens in G form a spanning tree of G (a spanning tree of a graph with n vertices has size $n-1$), and tokens are on all of the spikes of S (there are n spikes). This means that the $2n-1$ tokens must have been all moved to S in the way described above.

Now, restrict our view of the subgraph formed by the tokens to just those in S along with the edges cd, dv and call this H. As cdv is a path of length 2, S is the 1-spiked graph of G, and H is a caterpillar, by Lemma 3.2, we have that G contains a Hamiltonian path starting at v.

Now, let the size of G be $N+M$, where $N = |V(G)|$ and $M = |E(G)|$. Then the size of $G'_{cater}(v)$ is linear to $N+M$ since the size of P_1, P_2 is linear to N, Y is a constant, and the size of S is linear to $N+M$. Also, the size of E_s, E_t is linear to N. Then producing $(G'_{cater}(v), E_s, E_t)$ from G takes $O(N+M)$ time, which is polynomial w.r.t. the input size of G. So we have a polynomial-time reduction from the Hamiltonian v-path problem to the Caterpillar reconfiguration problem. □

3.2 k-bounded Path-width Tree Reconfiguration

Definition 3.5. Let T be a tree and v be a vertex of T, where $N(v) = \{v_1, \cdots, v_k\}$. Let $B_{v_1}, \cdots, B_{v_k}$ be the components of $T \setminus_V \{v\}$. We define B_{v_i} to be a *branch* of v in T with *root* v_i.

Definition 3.6. For $k \geq 0$, the k^{th} *layer* of a rooted tree, with root r, is the set of nodes that are at distance k away from r.

Definition 3.7. Let $k \geq 0$, $l \geq 1$. We define a (k,l)-*claw* C as follows:

1. Let T be a complete ternary tree of depth k with root r and leaf nodes $\{v_1, \cdots, v_{3^k}\}$.

2. For each leaf node v_i, we define a new path $P_{v_i} = p_1^{v_i} \cdots p_l^{v_i}$ of length l.

3. Then, C is the graph formed by $T \oplus_{v_i, p_l^{v_i}} P_{v_i}$ for all $i \in [3^k]$.

We say r is the *root* of C, T is the *core* of C, and each P_i a tail of C. Also, we call the branches of r in C, say B_1, B_2, B_3, the *root branches* of C, where each B_i contains path $\{P_{1+(i-1)\cdot 3^{k-1}}, \cdots, P_{i \cdot 3^{k-1}}\}$.

Furthermore, let $D(C)$ be a directed graph with the same vertex and edge set as C, with the root r as a source flowing outward until a leaf node is reached. Let $c, v \in V(C)$. We say c is a *child* of v and v is an *ancestor* of c if c is reachable from v in D.

Example 3.8. *A $(1,2)$-claw is the usual definition of a claw (i.e. a star with 3 edges).*

In order to prove our lemma about a claw's structure, we need to introduce the following structural theorem for path-width of trees.

Theorem 3.9 ([Sch90])**.** *Let $k \geq 1$ and T be a tree. $pw(T) \geq k+1$ if and only if there is a vertex $v \in V(T)$ such that there exists 3 branches, C_a, C_b, C_c, of v in T where $pw(C_a) \geq k$, $pw(C_b) \geq k$, and $pw(C_c) \geq k$.*

Proposition 3.10. *Let $k \geq 1$, G a connected graph, and v a vertex of G. Then, a (k,l)-claw has $\frac{3^{k+1}-1}{2} + 3^k l$ edges.*

Proof. There are $\sum_{i=1}^{k} 3^i$ edges in the ternary tree and 3^k paths with l edges, so we have $\sum_{i=1}^{k} 3^i + 3^k|E(P)| = \frac{3^{k+1}-1}{2} + 3^k l$ edges in total, as desired. □

Lemma 3.11. *Let $k \geq 1$. For any $l \geq 2$, a (k,l)-claw T with root r, core C, and tails P_i for $i \in [3^k]$ has the following properties:*

1. *$pw(T) = k+1$,*

2. *$pw(T \setminus_E P_i) = k$ for all $i \in [3^k]$.*

Moreover, we call each of $T \setminus_E P_i$ an almost (k,l)-claw without tail i, for all $i \in [3^k]$.

Proof Sketch. For part 1, we first note that a $(1,l)$-claw has path-width 2. Let C_1, C_2, C_3 be 3 copies of the $(1,l)$-claw with root r_1, r_2, r_3 respectively. Then, consider a normal claw G with central vertex r and leaf vertices a, b, c. Let $\tilde{G}$ be constructed by the following operations: $G \oplus_{a,r_1} C_1$, $G \oplus_{a,r_2} C_2$, $G \oplus_{a,r_3} C_3$. Note that $\tilde{G}$ is exactly a $(2,l)$-claw. By Theorem 3.9, G has path-width 3 since removal of any vertex v result in at most 1 branch with path-width 3, namely the branch containing an ancestor of v. The result follows inductively.

For part 2, the proof is similar to part 1. Note that an almost $(1,l)$-claw without tail P_1, P_2, or P_3, all have path-width 1. Then, using the $(1,l)$-claw as a base case and Theorem 3.9, we can show by induction on k that the branch of r in T containing P_i, say B_1, satisfies $pw(B_1) = k-2$, and the other two branches, say B_2, B_3, have path-width $k-1$. □

Lemma 3.12. *Let $k \geq 1, l \geq 2$. Let C be a (k,l)-claw. Let $Q = \{P_1, \cdots, P_{3^k}\}$ be the set of tails for C. Let T_i be trees with $t_i \in V(T_i)$ where $pw(T_i) \geq 2$ for $i \in [3^k]$. Then, if $G = C \oplus_{p_1^1, t_1} T_1 \oplus_{p_1^2, t_2} \cdots \oplus_{p_1^{3^k}, t_{3^k}} T_{3^k}$, then $pw(G) > k+1$.*

Proof. Let T_1, T_2, T_3 be trees with $pw(T_i) \geq 2$, for $i \in [3]$. Similar to before, consider joining (1-sum) the three trees with a $(1,l)$-claw using its tail's endpoints, and call this new graph G. Let r be the root of the $(1,l)$-claw. Then, the resulting graph has path-width w where $pw(T_i) + 1 \leq p \leq pw(T_i) + 2$, since a tail (excluding the root) joining the T_i for some $i \in [3]$ will at most increase $pw(T_i)$ by 1 (or remain the same). So, we have 3 branches of r each with path-width at least $pw(T_i) + 1 \geq 3$. So, $pw(G) \geq 3$. Iteratively building the claw from bottom-up, we have that each subtree of C at m^{th} layer of the core of C has $k - m + 2$. Since C is the subtree at layer 0, $pw(C) = k - 0 + 2 > k + 1$ as desired. □

Lemma 3.13. *Let $k \geq 1, l \geq 2$. Let C be an almost (k,l)-claw without tail j. Without loss of generality, assume $j = 3^k$. Let $C_1, \cdots, C_{3^k-1}$ be caterpillar graphs and v_i be an end of C_i for all $i \in [3^k - 1]$. Let G be a new graph. Then, if $G = C \oplus_{p_1^1, v_1} C_1 \oplus_{p_1^2, v_2} \cdots \oplus_{p_1^{3^k-1}, v_{3^k-1}} C_{3^k-1}$, then $pw(G) = k$.*

Proof Sketch. The proof should be very similar to that of Lemma 3.12 and part 2 of Lemma 3.11. Let T be an almost $(1,l)$-claw without tail 3. Let r be the root of T and P_1, P_2 be the tails of T. Let C_1, C_2 be caterpillar graphs with v_1, v_2 some end of C_1, C_2. Consider a the graph G by taking 1-sum of the two caterpillars' vertices v_1, v_2 with the $(1,l)$-claw tail endpoints p_1^1, p_2^2. Note that the graph $P_1 \oplus_{p_1^1, v_1} C_1$ has path-width 1, because the path P_1 and the spine of C_1 forms the new spine for G, where each edges is at most on distance away from the spine. Similarly for $P_2 \oplus_{p_1^2, v_2} C_2$. So, for each vertex $v \in V(G)$, there are at most 2 branches of v where the path-width is 1. By Theorem 3.9, $pw(G) = 1$. Then, using this G as a base case, and part 1 of Lemma 3.11, we can show inductively that, for each $v \in V(C)$, there are at most 2 branches of v with path-width k, so $pw(C) \leq k$. Since C contains a $(k-1,l)$-claw as a subgraph, by Lemma 3.11, $pw(C) = k$. □

Definition 3.14. Let $G = (V, E)$ be a non-empty graph where $V = \{v_1, \cdots, v_n\}$. Let v be a particular vertex of G. We define the auxiliary graph $G'_{path}(v)$ as follows:

- Let $g(k,n) = 3^{k-1}n^3$ and $f(k,n) = 3^{k-1} \cdot (n - 1 + n \cdot g(k,n)) + 2$. [1]
- Let T be a $(k, f(k,n))$-claw with root r, core C, tails P_i for $i \in [3^k]$, root branches B_1, B_2, B_3, and Q_1, Q_2, Q_3 be the tails contained within B_1, B_2, B_3 respectively.
- For all $1 \leq i \leq 3^{k-1}$, define a new graph G_i to a copy of G, where v_{G_i} of G_i corresponds to v of G. Then, let S_i be a $g(k,n)$-spiked graph of G_i with v_{S_i} in S_i corresponding to v_{G_i} of G_i.

Then, $G'_{path}(v)$ is formed by performing $T \oplus_{p_1^{2\cdot 3^{k-1}+i}, v_{S_i}} S_i$ for $i \in [3^{k-1}]$.

[1]The parameters are made somewhat loose for improved readability.

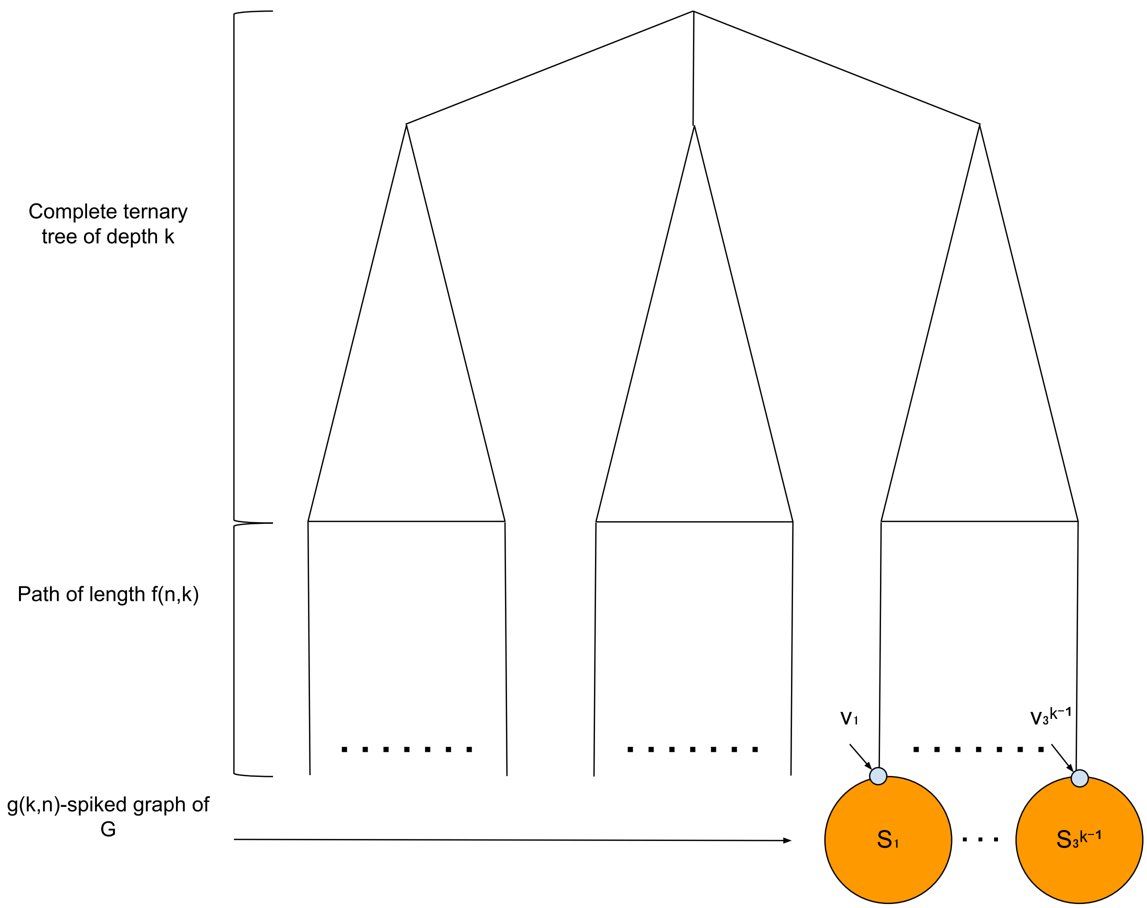


Figure 4: $G'_{path}(v)$

Theorem 3.15. *Let $k \geq K$ for some large enough constant K. The k-BOUNDED PATH-WIDTH TREE RECONFIGURATION problem under the TJ rule is* NP*-hard.*

Proof. To prove it is NP-hard, we show a polynomial-time reduction from the HAMILTONIAN v-PATH problem to the k-BOUNDED PATH-WIDTH TREE RECONFIGURATION problem. Let (G, v) be an instance of the HAMILTONIAN v-PATH problem, where G is non-empty. Using G, we create the auxiliary graph $G'_{path}(v)$ from 3.14. Let T be the $(k, f(k,n))$-claw in $G'_{path}(v)$ with root r, core C, tails P_i for $i \in [3^k]$, root branches B_1, B_2, B_3, and set of paths Q_1, Q_2, Q_3 restricted to within B_1, B_2, B_3 respectively. Let S_i be the spiked graphs in $G'_{path}(v)$ for $i \in [3^{k-1}]$. Let

$$E_s = E(T) \setminus E(P_{3^{k-1}+1}) \text{ and } E_t = E(T) \setminus E(P_1).$$

Then we want to show that $(G'_{path}(v), E_s, E_t)$ is reconfigurable if and only if G has a Hamiltonian path starting at v.

Claim 3.16. *Let $s \in \bigcup_{i=1}^{3^{k-1}} V(S_i)$. Let M_{total} be the total number of edges in all of S_i except the spikes at s, namely $M_{total} = (\sum_{i=1}^{3^{k-1}} |E(S_i)|) - g(k,n)$. Then, $M_{total} < f(k,n) - 2$.*

Proof. Note that, since $n \geq 1$,

$$\begin{aligned}(f(k,n)-2)-M &\geq [3^{k-1}\cdot(n-1+n\cdot g(k,n))]-[3^{k-1}\cdot(n\cdot g(k,n)+|E(K_n)|)-g(k,n)] \\ &= 3^{k-1}(n-1)-3^{k-1}\frac{n(n-1)}{2}+g(k,n) \\ &= 3^{k-1}(n^3-(\frac{n(n-1)}{2}-n+1))>0\end{aligned}$$

So, we have $f(k,n)-2 > M_{total}$ as desired. $\square$

($\Leftarrow$) Suppose that G has a Hamiltonian path starting at v, Let this Hamiltonian path be $e_1 \cdots e_{n-1}$, where e_1 is the edge starting at v. Then, for each $1 \leq i \leq 3^{k-1}$ Let $e^i_1 \cdots e^i_{n-1}$ denote the Hamiltonian path of G_i in S_i. Note that the number of edges on each Hamiltonian path is $n-1$. Also, note that the number of spikes for each S_i, for $i \in [3^{k-1}]$, is $n \cdot g(k,n) = n \cdot 3^k n^3$ and that each of the tail P_i, for $i \in [3^k]$, has $f(k,n)-1 = 3^{k-1}\cdot(n+n\cdot g(k,n))+1$ edges.

Then, one possible reconfiguration sequence from E_s to E_t in $G'_{path}(v)$ is:

(a) Let $i = 1$. For $i \leq 3^{k-1}$, we want to move $(n + n\cdot g(k,n))$ tokens to each S_i.
   (i) Let $a = i\cdot(n+n\cdot g(k,n))$.
   (ii) First, we move the token $p^1_{a+1}p^1_{a+2}$ to $e^{2\cdot 3^{k-1}+i}_1$, $\cdots$, $p^1_{a+n-1}p^1_{a+n}$ to $e^{2\cdot 3^{k-1}+i}_{n-1}$.
   (iii) Then, we move each of $p^1_{a+n}p^1_{a+(n+1)}, \cdots, p^1_{a+(n+n\cdot g(k,n)-1)}p^1_{a+(n+n\cdot g(k,n))}$ to each of the spikes in S_i sequentially.

(b) We move the token from $p^1_{f(k,n)-1}p^1_{f(k,n)}$ to $p^{3^{k-1}+1}_{f(k,n)-1}p^{3^{k-1}+1}_{f(k,n)}$.

(c) Finally, we symmetrically move all the tokens that we moved to each S_i in previous steps back to the tail $P_{3^{k-1}+1}$.

We claim that each step of reconfiguration sequence induces a graph that satisfies the property k-BOUNDED PATH-WIDTH TREE. It suffices to consider token jumps performed in step a of our reconfiguration sequence, since the remaining steps are symmetric in the sense that the graph induced is isomorphic to a graph induced by a previous reconfiguration step. Note that E_s induces a path-width k graph by part 2 of Lemma 3.11, since it is an almost (k,l)-claw.

Now, consider the j^{th} token jump in step a for some $1 \leq j \leq 3^{k-1}\cdot(n+n\cdot g(k,n))$, it induces a graph C_j what contains an almost $(k, f(k,n)-j)$-claw T_j without tail P_{3^k} as a subgraph. Furthermore, each tail P_i of T_j is augmented with a caterpillar graph G_i for $i \in [3^k-1]$, where each tail P_i is connected an end of G_i. From our reconfiguration sequence, it is clear each C_j maintains the tree property since C_j is just the 1-sum of a $(k, f(k,n)-j)$-claw and caterpillar graphs, which are both trees. Furthermore, note that $f(k,n)-j \geq 1$, since we only at most $f(k,n)-2$ edges away in step a. Then, by Lemma 3.13, $pw(C_j) = k$, so each subgraph induced in our reconfiguration sequence has path-width $\leq k$. Therefore, there exists a reconfiguration sequence from the source solution to the target solution, so $(G'_{path}(v), E_s, E_t)$ is reconfigurable.

($\Rightarrow$) Suppose that $(G'_{path}(v), E_s, E_t)$ is reconfigurable. Then, there exist a step during the reconfiguration in which a token is placed on the edge $p' = p^{3^{k-1}+1}_{f(k,n)}p^{3^{k-1}+1}_{f(k,n)-1}$. Note that each subgraph induced in our reconfiguration sequence has to be connected since the property of concern is a k-bounded path-width tree. Also, if tokens are placed on all edges of C and the first edge of each tail P_i (i.e. $p^i_{f(k,n)-1}p^i_{f(k,n)}$ for $i \in [3^k]$), then, by part 1 of Lemma 3.11, the graph induced by the edges has path-width $\geq k+1$ since it contains a subgraph isomorphic to a $(k, 2)$-claw.

So, if a token is placed on p' on step t, there exists at least one tail P_i for some $i \in [3^k] \setminus \{3^{k-1}+1\}$ such that no tokens are placed on any of its edges. This implies, at step $t-1$, right before moving the token to p', at least $f(k,n)-2$ tokens on P_i have already been moved to some other edges on $G'_{path}(v)$.

Without loss of generality, let the tail P_i described above, where there are no tokens placed on it at time t, to be the tail P_1. Due to the connectedness of tree, we cannot move any tokens to edges of $P_{3^{k-1}+1} \setminus_E \{p^{3^{k-1}+1}_{f(k,n)-1}p^{3^{k-1}+1}_{f(k,n)}\}$ before first moving a token to p'. So, the only place for the tokens to move to are the edges on each of the $g(k,n)$-spiked graph, namely S_i for $i \in [3^{k-1}]$. Since we have moved away at least $f(k,n)-2$ tokens of P_1 away at step t, by claim 3.16, there is at least a token on some spike at s for every vertex $s \in \bigcup_{i=1}^{3^{k-1}} V(S_i)$.

We make another observation. Let U be the subgraph induced by the tokens at step t. Similarly, for $i \in [3^{k-1}]$, let R_i denote the subgraph induced but constrained to the $g(k,n)$-spiked graph S_i and the tail $P_{2 \cdot 3^{k-1}+i}$. Note that U and all of the R_i's are connected subgraphs, and in particular, connected subtrees of $G'_{path}(v)$. Moreover, note that each S_i is connected to a leaf vertex in the root branch B_3, say with root v_3.

By a counting argument, U has to contain the edge rv_3, since a token can only be moved away from rv_3 when there are no tokens placed any of the edges of the root branches B_1, B_2. But at any point in the reconfiguration, there are at most $f(k,n) + (f(k,n) + g(k,n)) \leq 3f(k,n)$ slots to move to. For a large enough K with $k \geq K$, $|E(B_1)| + |E(B_3)| > 3f(k,n)$, since B_1 contains 3^k paths of length $f(k,n)$, namely the tails corresponding to B_1. So, $K = 2$ suffices. Since there is a token edge rv_3 and there are tokens on each S_i, there are tokens on all of the edges in B_2. So, U contains B_2 as a subgraph.

Now, suppose, for a contradiction, that $pw(R_i) \geq 2$ for all $i \in [3^{k-1}]$. Then, by Lemma 3.12, we can show that $pw(U) > k$, since U contains a $(k-1, 2)$-claw subgraph where for $i \in [3^{k-1}]$ each tail P_i are 1-summed with R_i (which are of trees with path-width ≥ 2). This is a contradiction since step t is part of the reconfiguration sequence. So, there exists some $i \in [3^{k-1}]$ where R_i is a path-width 1 tree (i.e. a caterpillar), and without loss of generality, we assume $R_i = R_1$.

Note that R_1 is a subgraph of the 1-sum of path P_1 and a $g(k,n)$-spiked graph. Also, by above observation, there is at least a token on some spike at s for every vertex $s \in \bigcup_{i=1}^{3^{k-1}} V(S_1)$, so by connectedness of R_1, the R_1 has to contain a spanning tree of G_1 in S_1, and contain at least one spike for each vertex of G_1. So, by Lemma 3.2, there is a Hamiltonian $v_{S_1} = v_{G_1}$ path on G_1. Since G_1 is just a copy of G and $v_{G_1} = v$, there is a Hamiltonian v-path for G as desired.

Now, let the size of G be $N+M$, where $N = |V(G)|$ and $M = |E(G)|$. Since $M < N^2$, it suffices to show that our input $(G'_{path}(v), E_s, E_t)$ has size polynomial to N. By the construction of $G'_{path}(v)$, it contains a $(k, f(k,n))$-claw subgraph and 3^{k-1} $g(k,N)$-spiked graphs of G. By proposition 3.10, the $(k, f(k,n))$-claw subgraph has $\frac{3^{k+1}-1}{2} + 3^k \cdot f(k,N)$ edges, thus $\frac{3^{k+1}-1}{2} + 3^k \cdot f(k,N) + 1$ vertices. The spiked graphs have $3^{k-1} \cdot (N \cdot g(k,N) + N)$ vertices in total. Recall $g(k,N) = 3^{k-1}N^3$ and $f(k,N) = 3^{k-1} \cdot (N - 1 + N \cdot g(k,N)) + 2$. Then,

$$\begin{aligned}
|V(G'_{path}(v))| &= \frac{3^{k+1}-1}{2} + 3^k \cdot f(k,N) + 1 + 3^{k-1} \cdot (N \cdot g(k,N) + N) \\
&= \frac{3^{k+1}-1}{2} + 3^k \cdot (3^{k-1} \cdot (N - 1 + N \cdot 3^{k-1}N^3) + 2) + 1 + 3^{k-1} \cdot (N \cdot 3^{k-1}N^3 + N) \\
&\leq 3^{k+1} + 3^{2k}N + 3^{3k}N^4 + 3^{k+1} + 2 \cdot 3^k + 1 + 3^{2k}N^4 + 3^kN \\
&\leq 3^{3k}N^4 \leq 3^{3k+2}N^4.
\end{aligned}$$

Since k is a constant (k is fixed), the size of $G'_{path}(v) = |V(G'_{path}(v))| + |E(G'_{path}(v))| \leq 3^{3k+2}N^4 + (3^{3k+2}N^4)^2$, which is polynomial in N. Also, note that the size of E_s, E_t is bounded above by $|E(G'_{path}(v))| < |V(G'_{path}(v))|^2 \leq 3^{6k+4}N^8$, so they are also polynomial in N. So, producing our input takes $O(poly(N + M))$ time, which is polynomial w.r.t. the input size of G. So we have a polynomial-time reduction as desired. □

4 Conclusion and Open Problems

In this paper, we have proved that the Caterpillar reconfiguration problem is NP-hard under the TJ rule, and attempted to prove this for the k-bounded path-width tree reconfiguration as well. Some open problems are the complexities for our two studied properties but under the TS rule. Results for trees and paths under the TS rule are also still unknown. Understanding their behavior would likely provide insights into how the bounded path-width and bounded tree-width versions behave under the TS rule. While we suspect k-bounded path-width reconfiguration is in NP, we are not aware of any formal proof of this result.

Acknowledgments

Jeffrey Kam would like to thank Prof. Naomi Nishimura for helpful discussions and feedbacks.

References

[DEH+19] Erik D. Demaine, David Eppstein, Adam Hesterberg, Kshitij Jain, Anna Lubiw, Ryuhei Uehara, and Yushi Uno. Reconfiguring undirected paths. In Zachary Friggstad, Jörg-Rüdiger Sack, and Mohammad R Salavatipour, editors, *Algorithms and Data Structures*, pages 353–365, Cham, 2019. Springer International Publishing.

[HIM+20] Tesshu Hanaka, Takehiro Ito, Haruka Mizuta, Benjamin Moore, Naomi Nishimura, Vijay Subramanya, Akira Suzuki, and Krishna Vaidyanathan. Reconfiguring spanning and induced subgraphs. *Theoretical Computer Science*, 806:553–566, 2020.

[RS83] Neil Robertson and P.D. Seymour. Graph minors. i. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983.

[Sch90] Petra Scheffler. A linear algorithm for the pathwidth of trees. In *Topics in combinatorics and graph theory*, pages 613–620. Springer, 1990.