

Shortest Reconfiguration Sequence for Sliding Tokens on Spiders

Duc A. Hoang^{1, 3} Amanj Khorramian² Ryuhei Uehara¹

May 12–17, 2019

¹School of Information Science, JAIST, Japan

²University of Kurdistan, Sanandaj, Iran

³Kyushu Institute of Technology, Japan [As of April 01, 2019]

The SHORTEST SLIDING TOKEN problem

The SLIDING TOKEN problem

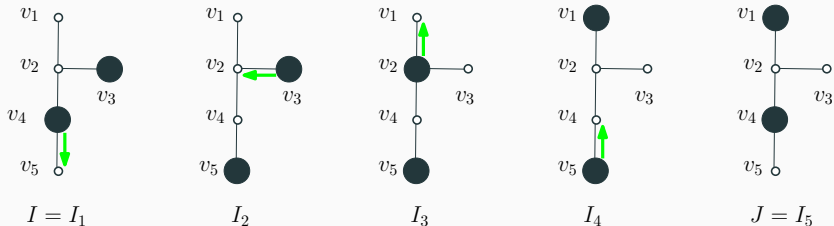
SLIDING TOKEN [Hearn and Demaine 2005]

Given

two independent sets (token sets) I, J of a graph G , and the Token Sliding (TS) rule

Ask

whether there is a TS-sequence that transforms I into J (and vice versa)



A TS-sequence that transforms $I = I_1$ into $J = I_5$. Vertices of an independent set are marked with black circles (tokens).

The SHORTEST SLIDING TOKEN problem

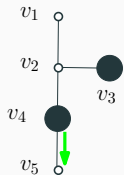
SHORTEST SLIDING TOKEN [Yamada and Uehara 2016]

Given

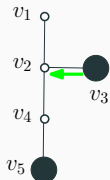
a yes-instance (G, I, J) of SLIDING TOKEN, where I, J are independent sets of a graph G

Ask

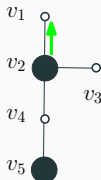
find a shortest TS-sequence that transforms I into J (and vice versa)



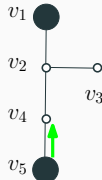
$I = I_1$



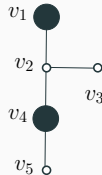
I_2



I_3



I_4



$J = I_5$

A shortest TS-sequence that transforms $I = I_1$ into $J = I_5$. Vertices of an independent set are marked with black circles (tokens).

Theorem (Kamiński et al. 2012)

It is NP-complete to decide if there is a TS-sequence having at most ℓ token-slides between two independent sets I, J of a perfect graph G even when ℓ is polynomial in $|V(G)|$.

Theorem (Kamiński et al. 2012)

SHORTEST SLIDING TOKEN can be solved in linear time for cographs (P_4 -free graphs).

Theorem (Yamada and Uehara 2016)

SHORTEST SLIDING TOKEN can be solved in polynomial time for proper interval graphs, trivially perfect graphs, and caterpillars.

Very recently, it has been announced that

Theorem (Sugimori, AAAC 2018)

SHORTEST SLIDING TOKEN *can be solved in $O(\text{poly}(n))$ time when the input graph is a tree T on n vertices.*

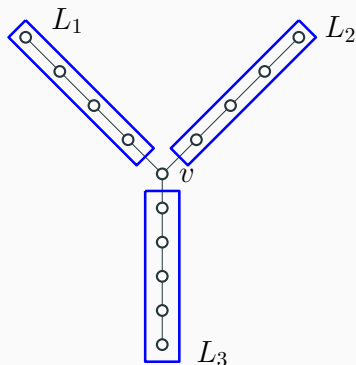
- Sugimori's algorithm uses a dynamic programming approach. (We are still waiting for a formal version of his algorithm.)
- The order of $\text{poly}(n)$ seems to be large.

Theorem (Our Result)

SHORTEST SLIDING TOKEN *can be solved in $O(n^2)$ time when the input graph is a spider G (i.e., a tree having exactly one vertex of degree at least 3) on n vertices.*

- We hope that our algorithm provides some ideas for improving the running time of Sugimori's algorithm.

SHORTEST SLIDING TOKEN **for** **Spiders**

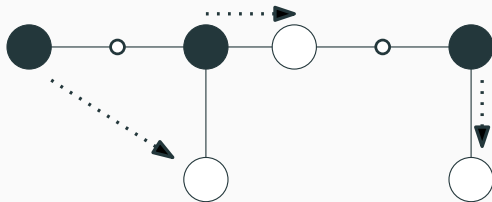


A spider graph

A **spider** G is specified in terms of

- a **body** vertex v whose degree is at least 3; and
- $d = \deg_G(v)$ **legs** L_1, L_2, \dots, L_d attached to v

- For caterpillars, only **one way** to move tokens.
 - Rightmost black token \rightarrow rightmost white token

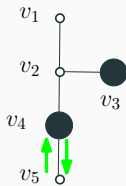


- In general, there are $k!$ **possible ways** of assigning targets for k tokens.

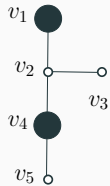
Challenge

Knowing **where to move tokens to**.

- When a token is placed at its “destination”, but needs to move away so that another token can move.

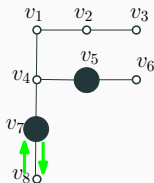


I

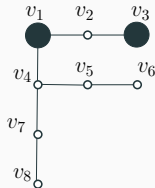


J

- When a token needs to move “far away” from its “destination” so that another token can move.



I



J

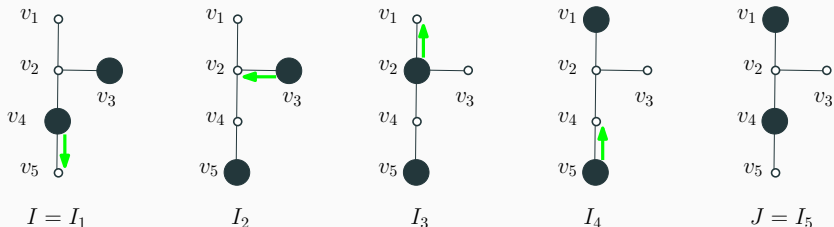
Challenge

Knowing [when and how to make detours](#).

- Both **Token Sliding** and **Token Swapping** describe “local changes” .
- Finding a reconfiguration sequence is **easy** (can be done in polynomial time).
- Finding a **minimum** reconfiguration sequence is **difficult** because of the challenge when handling a special structure (“detour” in **Token Sliding**, and “happy leaf” in **Token Swapping**).

Detour: A Formal Definition

We say that a TS-sequence S makes detour over an edge $e = xy \in E(G)$ if S at some time moves a token from x to y , and at some other time moves a token from y to x .



S makes detour over $e = v_4v_5$

The body vertex v is crucial. Roughly speaking, we explicitly construct a shortest TS-sequence when

- **Case 1:** $\max\{|I \cap N_G(v)|, |J \cap N_G(v)|\} = 0$
 - No token is in the neighbor $N_G(v)$ of v
 - Detour is **not** required
- **Case 2:** $0 < \max\{|I \cap N_G(v)|, |J \cap N_G(v)|\} \leq 1$
 - At most one token (from either I or J) is in the neighbor $N_G(v)$ of v
 - Detour is **sometimes** required
- **Case 3:** $\max\{|I \cap N_G(v)|, |J \cap N_G(v)|\} \geq 2$
 - At least two tokens (from either I or J) are in the neighbor $N_G(v)$ of v
 - Detour is **always** required

A **target assignment** is simply a bijective mapping $f : I \rightarrow J$.

Observe that

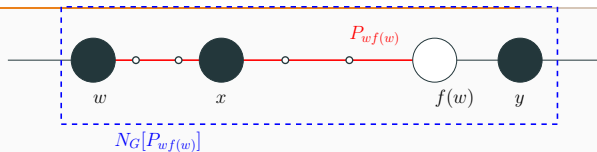
- Any TS-sequence S induces a target assignment f_S .
- Thus, each S uses at least $\sum_{w \in I} \text{dist}_G(w, f_S(w))$ token-slides.

Indeed,

Lemma (Key Lemma)

One can construct in linear time a target assignment f that minimizes $\sum_{w \in I} \text{dist}_G(w, f(w))$, where $\text{dist}_G(x, y)$ denotes the distance between two vertices x, y of a spider G .

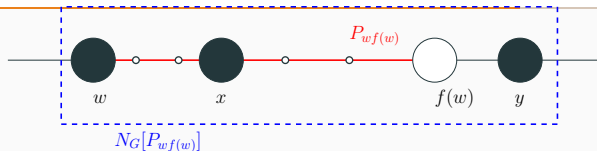
Case 1: $\max\{|I \cap N_G(v)|, |J \cap N_G(v)|\} = 0$



Observation

In the figure above, w can be moved to $f(w)$ along the shortest path $P_{wf(w)}$ between them only **after** both x and y are moved.

Case 1: $\max\{|I \cap N_G(v)|, |J \cap N_G(v)|\} = 0$



Observation

In the figure above, w can be moved to $f(w)$ along the shortest path $P_{wf(w)}$ between them only **after** both x and y are moved.

Theorem

When $\max\{|I \cap N_G(v)|, |J \cap N_G(v)|\} = 0$, one can construct a (shortest) TS-sequence using M^* token-slides between I and J , where $M^* = \min_{\text{target assignment } f} \sum_{w \in I} \text{dist}_G(w, f(w))$. Moreover, this construction takes $O(|V(G)|^2)$ time.

Proof Sketch: The Key Lemma allows us to **pick a “good” target assignment**, and the above observation tells us **which token should be moved first**.

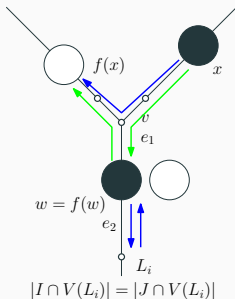
Case 2: $\max\{|I \cap N_G(v)|, |J \cap N_G(v)|\} \leq 1$

Special Case

- w and $f(w)$ are both in $N_G(v) \cap V(L_i)$;
- the number of I -tokens and J -tokens in L_i are equal.

In this case, any TS-sequence must (at least) make detour over either e_1 or e_2 .

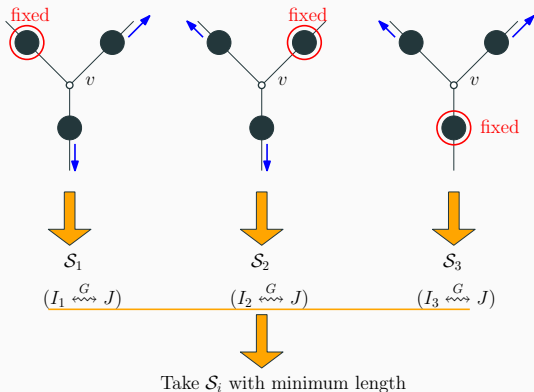
- To handle this case, simply move both w and $f(w)$ to v . The problem now reduces to **Case 1**.
- This is **not true** when each leg of G contains the same number of I -tokens and J -tokens. However, this case is easy and can be handled separately.
- When the above case does not happen, slightly modify the instance to reduce to **Case 1**.



Case 3: $\max\{|I \cap N_G(v)|, |J \cap N_G(v)|\} \geq 2$

We consider only the case $|I \cap N_G(v)| \geq 2$ and $|J \cap N_G(v)| \leq 1$.

Other cases are similar.



- For any TS-sequence S , **exactly one of the $d = \deg_G(v)$ situations (as in the above example) must happen.**
- Applying the above trick (regardless of J -tokens) reduces the problem to known cases (either **Case 1** or **Case 2**).

Conclusion

- We provided a $O(n^2)$ -time algorithm for solving SHORTEST SLIDING TOKEN for spiders on n vertices.
- A shortest TS-sequence is explicitly constructed, along with the number of detours it makes.
- Our algorithm is optimal in the number of token-slides, as there exists a shortest TS-sequence having $\Omega(n^2)$ token-slides (see Demaine et al. 2015).

Future Work

- Extend the framework to improve the running time of Sugimori's algorithm for trees.
- What about the graphs containing cycles?



Demaine, Erik D., Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada (2015). "Linear-time algorithm for sliding tokens on trees". In: *Theoretical Computer Science* 600, pp. 132–142. DOI: 10.1016/j.tcs.2015.07.037.



Hearn, Robert A. and Erik D. Demaine (2005). "PSPACE-Completeness of Sliding-Block Puzzles and Other Problems through the Nondeterministic Constraint Logic Model of Computation". In: *Theoretical Computer Science* 343.1-2, pp. 72–96. DOI: 10.1016/j.tcs.2005.05.008.



Kamiński, Marcin, Paul Medvedev, and Martin Milanič (2012). "Complexity of independent set reconfigurability problems". In: *Theoretical Computer Science* 439, pp. 9–15. DOI: 10.1016/j.tcs.2012.03.004.



Yamada, Takeshi and Ryuhei Uehara (2016). "Shortest reconfiguration of sliding tokens on a caterpillar". In: *Proceedings of WALCOM 2016*. Ed. by Mohammad Kaykobad and Rossella Petreschi. Vol. 9627. LNCS. Springer, pp. 236–248. DOI: 10.1007/978-3-319-30139-6_19.