Polynomial-Time Algorithm for Sliding Tokens on Trees

Erik D. Demaine[1], Martin L. Demaine[1], Eli Fox-Epstein[2], Duc A. Hoang[3], Takehiro Ito[4], Hirotaka Ono[5], Yota Otachi[3], Ryuhei Uehara[3], and Takeshi Yamada[3]

[1]Massachusetts Institute of Technology, USA. {edemaine, mdemaine}@mit.edu
[2]Brown University, USA. ef@cs.brown.edu
[3]JAIST, Japan. {hoanganhduc, otachi, uehara, tyama}@jaist.ac.jp
[4]Tohoku University, Japan. takehiro@ecei.tohoku.ac.jp
[5]Kyushu University, Japan. hirotaka@econ.kyushu-u.ac.jp

The *reconfiguration problem* arises when we would like to know whether one can transform one feasible solution of a problem into another by some allowable set of moves. Many important variants of this problem have been considered recently, including the *independent set reconfiguration problem*. A recent survey by van den Heuvel [5] gives a good introduction to this research area.

Given a graph G with n vertices, m edges, and two independent sets $\mathbf{I}_b$ and $\mathbf{I}_r$ of G, imagine that a token is put at each vertex of $\mathbf{I}_b$. The independent set reconfiguration problem ISReconf asks whether we can transform $\mathbf{I}_b$ to $\mathbf{I}_r$ via a sequence of independent sets of G, each of which results from the previous one by moving a token under some given *reconfiguration rules*, namely token sliding (TS), token jumping (TJ), and token addition and removal (TAR). The ISReconf problem under TS rule, in which tokens may only be moved to adjacent vertices, is called the sliding token problem and is of particular theoretical interest. This sliding token problem is known to be PSPACE-complete even for planar graphs. Ito et al. [3] proved that ISReconf under TAR rule is PSPACE-complete for general graphs. Kamiński et al. [4] proved that ISReconf is PSPACE-complete for perfect graphs under any of three reconfiguration rules. Additionally, they gave an $O(n+m)$-time algorithm to solve the sliding token problem for cographs. Very recently, Bonsma et al. [1] proved that sliding token can be solved in polynomial time for claw-free graphs. As the independent set problem plays an important role in computational complexity theory, it is important to resolve the complexity of ISReconf on some critical graph classes.

In this extended abstract, we give a brief description of our recent result on the sliding token problem for trees [2]. For more details on the notations and concepts, refer to the original paper [2].

Let T be a tree with two independent sets $\mathbf{I}_b$ and $\mathbf{I}_r$ of equal cardinality, and imagine that a token is placed on each vertex of $\mathbf{I}_b$. The sliding token problem asks whether there exists a sequence $\langle \mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_\ell \rangle$ of independent sets of T such that:

(i) $\mathbf{I}_1 = \mathbf{I}_b$, $\mathbf{I}_\ell = \mathbf{I}_r$, and $|\mathbf{I}_i| = |\mathbf{I}_b| = |\mathbf{I}_r|$ for all i, $1 \le i \le \ell$; and

(ii) There is an edge uv of T such that $\mathbf{I}_i$ can be obtained from $\mathbf{I}_{i-1}$ by sliding a token from vertex $u \in \mathbf{I}_{i-1}$ to its adjacent vertex v along the edge uv.

Theorem 1. *The sliding token problem can be solved in $O(n^2)$ time for any tree T with n vertices.*

Let T be a tree and let $\mathbf{I}$ be an independent set of T. For two distinct vertices u and v of a tree T, let T^u_v be the subtree of T obtained by regarding u as the root of T and then taking the subtree rooted at v which consists of v and all descendants of v (see Figure 1). It should be noted that u is not contained in the subtree T^u_v.

We say that a token on a vertex $v \in \mathbf{I}$ is $(T, \mathbf{I})$-*rigid* if $v \in \mathbf{I}'$ holds for any independent set $\mathbf{I}'$ of T such that $\mathbf{I} \overset{T}{\leftrightsquigarrow} \mathbf{I}'$, where $\mathbf{I} \overset{T}{\leftrightsquigarrow} \mathbf{I}'$ indicates that we can transform $\mathbf{I}$ to $\mathbf{I}'$ via sliding tokens on T. A token on $v \in \mathbf{I}$ is $(T, \mathbf{I})$-*movable* if it is not $(T, \mathbf{I})$-rigid. Similarly, we can define the concept of rigid/movable for a

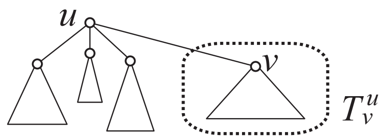

Figure 1: Subtree T_v^u in the whole tree T.

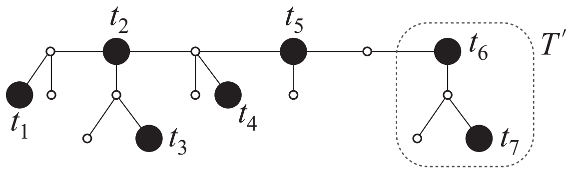

Figure 2: An independent set $\mathbf{I}$ of a tree T, where t_1, t_2, t_3, t_4 are $(T, \mathbf{I})$-rigid tokens and t_5, t_6, t_7 are $(T, \mathbf{I})$-movable tokens. For the subtree T', tokens t_6, t_7 are $(T', \mathbf{I} \cap T')$-rigid.

subtree T' of T. For a subtree T' of T, denote by $\mathbf{I} \cap T'$ the set $\mathbf{I} \cap V(T')$. A token on a vertex $v \in \mathbf{I} \cap T'$ is $(T', \mathbf{I} \cap T')$-*rigid* if $v \in \mathbf{J}$ holds for any independent set $\mathbf{J}$ of T' such that $\mathbf{I} \cap T' \overset{T'}{\leftrightsquigarrow} \mathbf{J}$. Figure 2 gives an illustration for these definitions. An important recursive characterization of rigid tokens is as follows:

Lemma 1. *Let $\mathbf{I}$ be an independent set of a tree T, and let u be a vertex in $\mathbf{I}$. Denote by $N(T, u)$ the set of all neighbors of u in T.*

(*a*) *Suppose that $|V(T)| = |\{u\}| = 1$. Then, the token on u is $(T, \mathbf{I})$-rigid.*

(*b*) *Suppose that $|V(T)| \geq 2$. Then, a token on u is $(T, \mathbf{I})$-rigid if and only if, for all neighbors $v \in N(T, u)$, there exists a vertex $w \in \mathbf{I} \cap N(T_v^u, v)$ such that the token on w is $(T_w^v, \mathbf{I} \cap T_w^v)$-rigid.*

From Lemma 1, we claim the following lemma.

Lemma 2. *It can be decided in $O(n)$ time whether the token on u is $(T, \mathbf{I})$-rigid.*

Next, we describe our algorithm to solve the SLIDING TOKEN problem on trees. Denote by $\mathsf{R}(\mathbf{I})$ the set of all vertices in $\mathbf{I}$ on which $(T, \mathbf{I})$-rigid tokens are placed. Let $N[T, \mathsf{R}(\mathbf{I})] = \bigcup_{v \in R(\mathbf{I})} \big(N(T, v) \cup v\big)$.

Step 1. Compute $\mathsf{R}(\mathbf{I}_b)$ and $\mathsf{R}(\mathbf{I}_r)$ using Lemma 2. If $\mathsf{R}(\mathbf{I}_b) \neq \mathsf{R}(\mathbf{I}_r)$, then return "no"; otherwise go to Step 2.

Step 2. Delete the vertices in $N[T, \mathsf{R}(\mathbf{I}_b)] = N[T, \mathsf{R}(\mathbf{I}_r)]$ from T, and obtain a forest F consisting of q trees $T_1, T_2, \ldots, T_q$. If $|\mathbf{I}_b \cap T_j| = |\mathbf{I}_r \cap T_j|$ holds for every $j \in \{1, 2, \ldots, q\}$, then return "yes"; otherwise return "no."

By Lemma 2 we can determine whether one token in an independent set $\mathbf{I}$ of T is $(T, \mathbf{I})$-rigid or not in $O(n)$ time, and hence Step 1 can be done in time $O(n) \times (|\mathbf{I}_b| + |\mathbf{I}_r|) = O(n^2)$. Clearly, Step 2 can be done in $O(n)$ time. Therefore, our algorithm above runs in $O(n^2)$ time in total.

References

[1] Paul Bonsma, Marcin Kamiński, and Marcin Wrochna. Reconfiguring independent sets in claw-free graphs. *arXiv:1403.0359*, March 2014.

[2] Erik D. Demaine, Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada. Polynomial-time algorithm for sliding tokens on trees. *arXiv:1406.6576*, June 2014.

[3] Takehiro Ito, Erik D. Demaine, Nicholas J.A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12-14):1054–1065, March 2011.

[4] Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theoretical Computer Science*, 439:9–15, June 2012.

[5] Jan van den Heuvel. The complexity of change. *arXiv:1312.2816*, December 2013.