The Application of Forbidden Structures in Solving Reconfiguration Problems

Duc A. Hoang

Graduate School of Informatics, Kyoto University

Reconfiguration Problems

... study the relationship between *solutions* of a given *source problem* (e.g., SATISFIABILITY, INDEPENDENT SET, VERTEX COVER, VERTEX-COLORING, etc.).

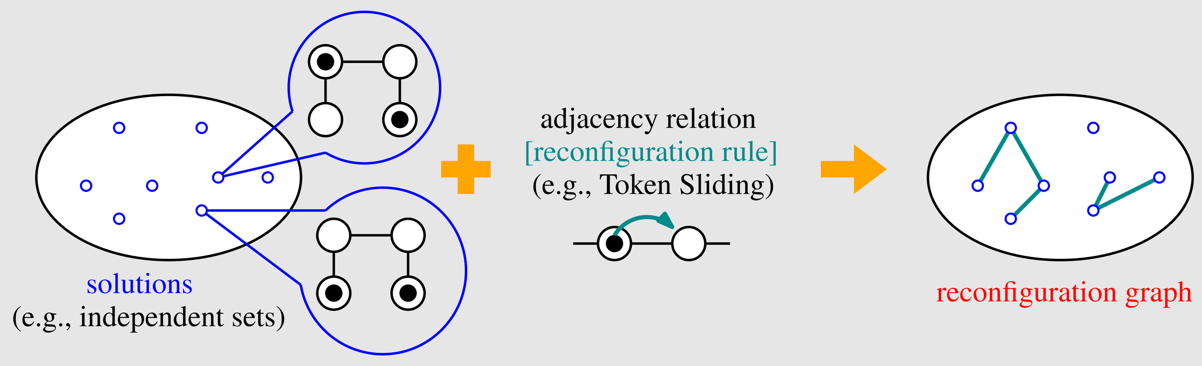

Figure 1. Reconfiguration.

Some typical questions:

- REACHABILITY: Is there a *path* between two given solutions?
- SHORTEST TRANSFORMATION: If REACHABILITY is yes, can we find a *shortest path*?
- CONNECTIVITY: Is there a *path* between *any* two given solutions?
- DIAMETER: Is the *maximum distance* between any two solutions bounded?
- and so on.

Example Problem: Reconfiguration of p-PATH VERTEX COVER

- Source Problem: p-PATH VERTEX COVER ($p \geq 2$).
  - **Input:** $G = (V, E)$, k.
  - **Question:** Is there a *p-path vertex cover (p-PVC)* of G, i.e., a vertex-subset I s.t. every path on p vertices in G contains at least one vertex from I, of size at most k?
  - A *2-PVC* I is also called a *vertex cover* and $V \setminus I$ is an *independent set*.
- Adjacency Relation: Token Sliding (TS).
  - Two p-PVCs I, J are *adjacent* if there exist $u, v \in V$ s.t. $I \setminus J = \{u\}$, $J \setminus I = \{v\}$, and $uv \in E$.
- Reconfiguration Graph: $\mathsf{TS}^p_k(G)$ and $\mathsf{TS}^p(G)$.

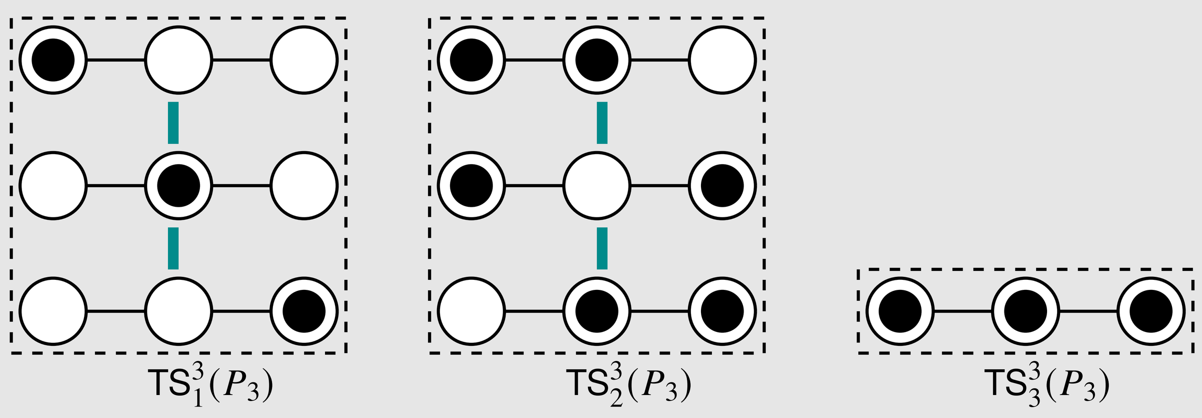

Figure 2. $\mathsf{TS}^3(P_3)$.

Note: Reconfiguration graphs for VERTEX COVER and INDEPENDENT SET are the same. When $p = 2$, since several known results use INDEPENDENT SET as the source problem, we use it instead of VERTEX COVER, and use $\mathsf{TS}_k(G)$ and $\mathsf{TS}(G)$ to indicate $\mathsf{TS}^2_k(G)$ and $\mathsf{TS}^2(G)$, respectively.

Forbidden Structures in Reconfiguration

- A *forbidden structure* is *part of a solution* S satisfying certain properties that *obstruct the existence of a path* in the reconfiguration graph between S and some other solution.
- Example: *rigid tokens* (= tokens that never move) in each node of $\mathsf{TS}_k(G)$.

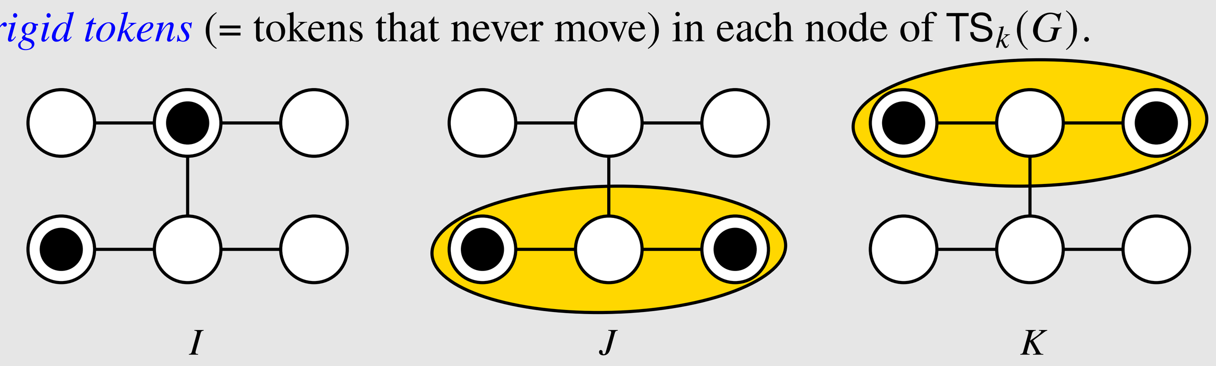

Figure 3. Independent sets $I, J, K \in \mathsf{TS}_2(G)$ having no path connecting any two of them.

- *Naturally, in $\mathsf{TS}^p(G)$, forbidden structures involve tokens whose movements are "restricted".*

REACHABILITY in $\mathsf{TS}(G)$

- *Polynomial-time algorithms* when G is a *tree* [Demaine et al. 2015], *bipartite permutation graph* [Fox-Epstein et al. 2015], or *cactus graph* [Hoang and Uehara 2016].
- **Open Problem:** G is *outerplanar*?

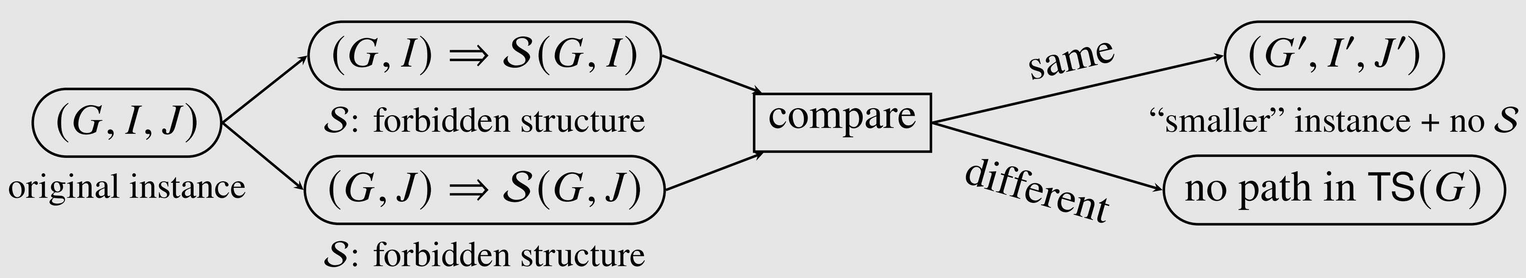

Figure 4. A general approach to solve REACHABILITY using forbidden structures.

REACHABILITY in $\mathsf{TS}^p(G)$ ($p \geq 3$)

- PSPACE-*hardness* when G is *planar* $\cap$ *maximum degree 3* [Hoang, Suzuki, and Yagita 2020].
- AND/OR NCL graph.
  - Blue edge $\Rightarrow$ weight 2.
  - Red edge $\Rightarrow$ weight 1.
  - Total in-weight ≥ 2.
- Reverse edge-direction.
- Each dashed rectangle represents a connecting part.
- *Design gadgets s.t. no token ever leaves its connecting part.*
- **Open Problem:** G is *bipartite*?

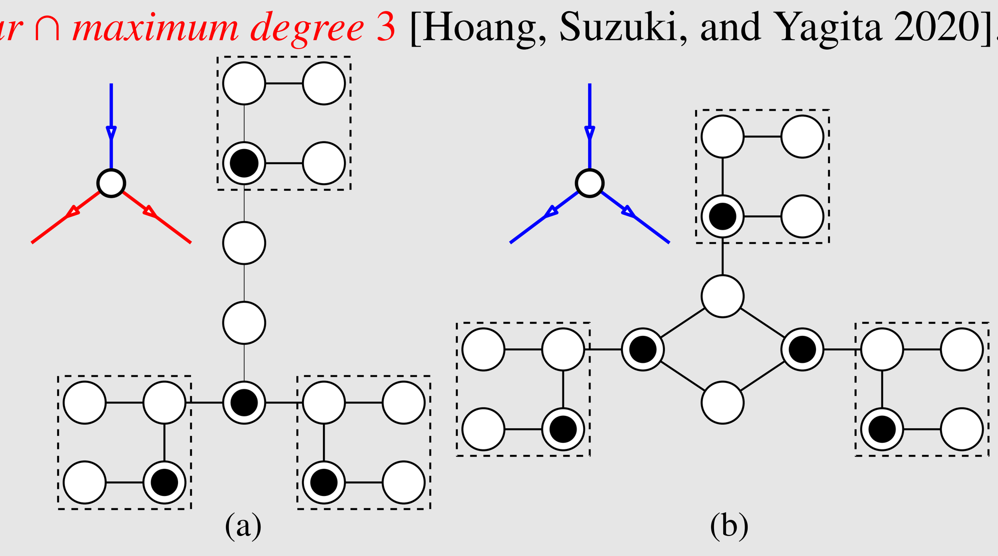

Figure 5. Gadgets when $p = 3$. (a) AND gadget. (b) OR gadget.

Find G s.t. $\mathsf{TS}_k(G)$ ($k \geq 2$) Has Certain Properties [Avis and Hoang 2022]

- *Cliques.* (G contains $K_n \Leftrightarrow \mathsf{TS}(G)$ contains K_n.)
  There is G s.t. G has a K_n and $\mathsf{TS}_k(G)$ ($k \geq 2$) doesn't.
  Design G s.t. exactly one token moves inside a clique.

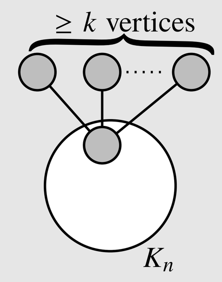

Figure 6. A desired graph G.

- *Eulerianity.*

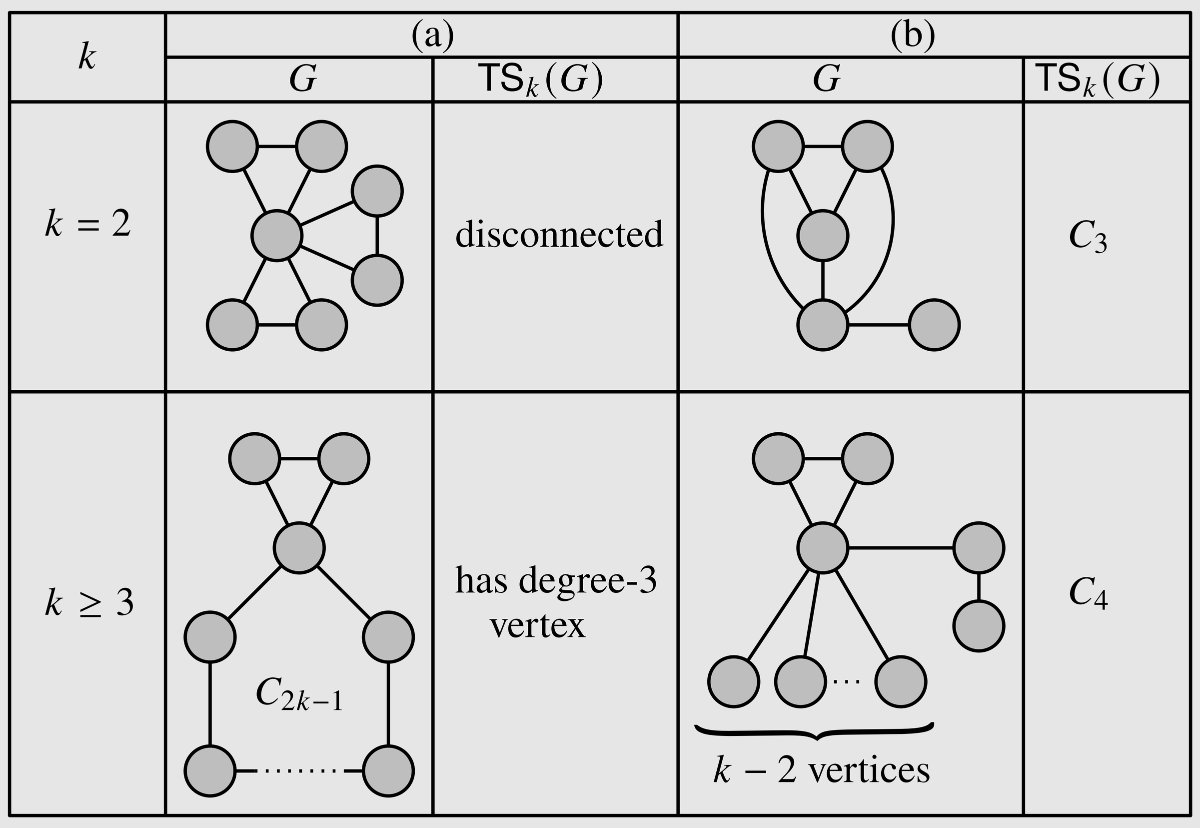

Figure 7. (a) G is Eulerian and $\mathsf{TS}_k(G)$ isn't. (b) G isn't Eulerian and $\mathsf{TS}_k(G)$ is.

Design G s.t. some special token has exactly one way to move.

- **Future Goal:** More graph properties?

Find H s.t. $\mathsf{TS}_k(H) \simeq G$ ($k \geq 2$) for some graph G [Avis and Hoang 2022]

- $G = P_n$ and $k = 2$: *Design H s.t. once a token is on a_{n+1}, the only choice for placing the other is on a_n.* ($V(H) = \{a_1, \dots, a_n, a_{n+1}\}$.)
- Generalize for fixed $k \geq 3$: Join every vertex of $H \simeq \mathsf{TS}_2(P_n)$ to the center of a star $K_{1,k-2}$. Note that the *$k - 2$ tokens on the star never move.* (H has no size-3 independent set.)

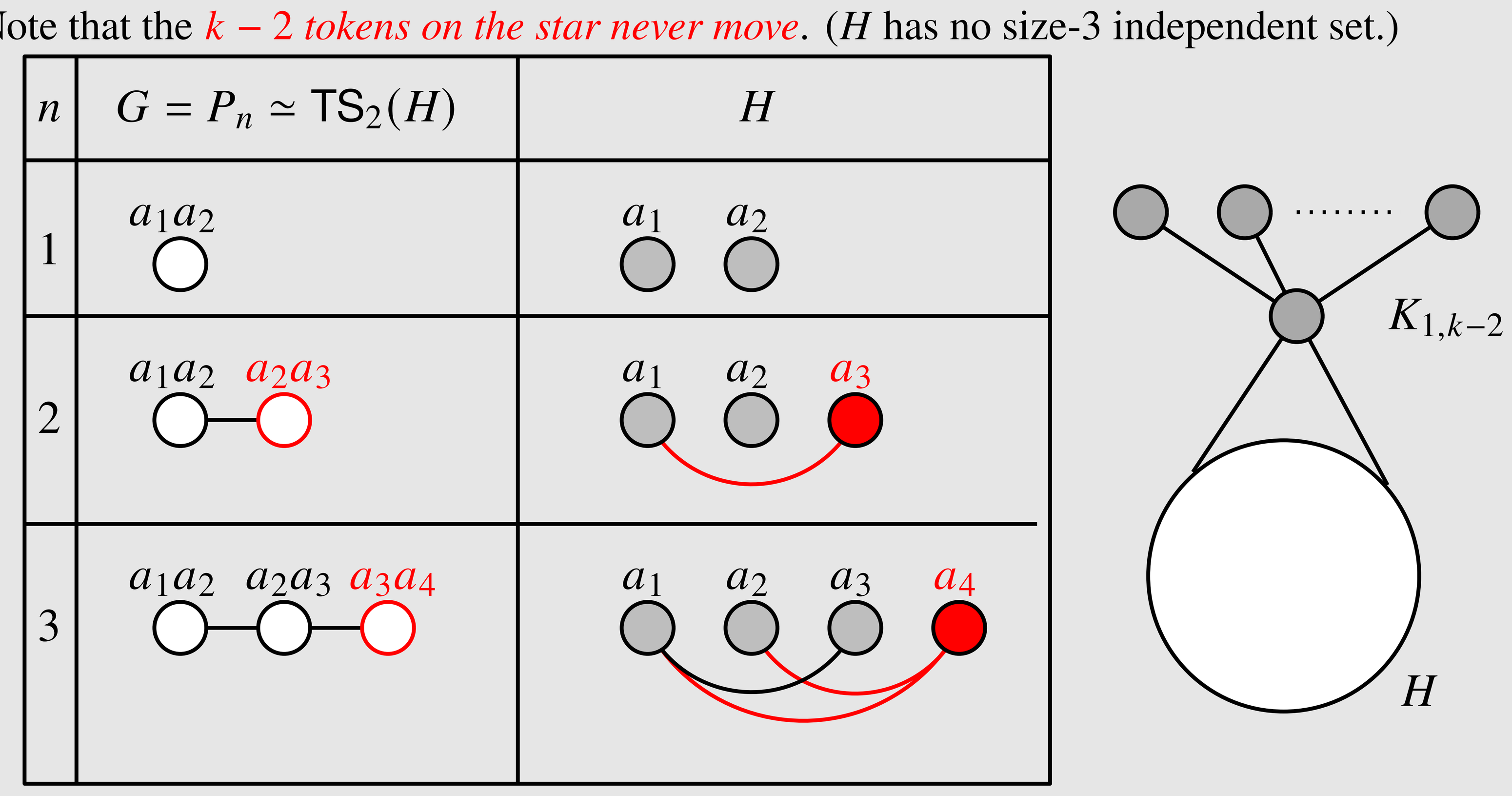

Figure 8. Design H s.t. $\mathsf{TS}_k(H) \simeq G$ ($k \geq 2$) where $G = P_n$.

- $G = K_{1,n}$ ($n \leq k$). (If $n > k$, there is no H s.t. $\mathsf{TS}_k(H) \simeq K_{1,n}$.) *Design H s.t. each token on a_i ($1 \leq i \leq k$) either moves back-and-forth along an edge or never moves.*

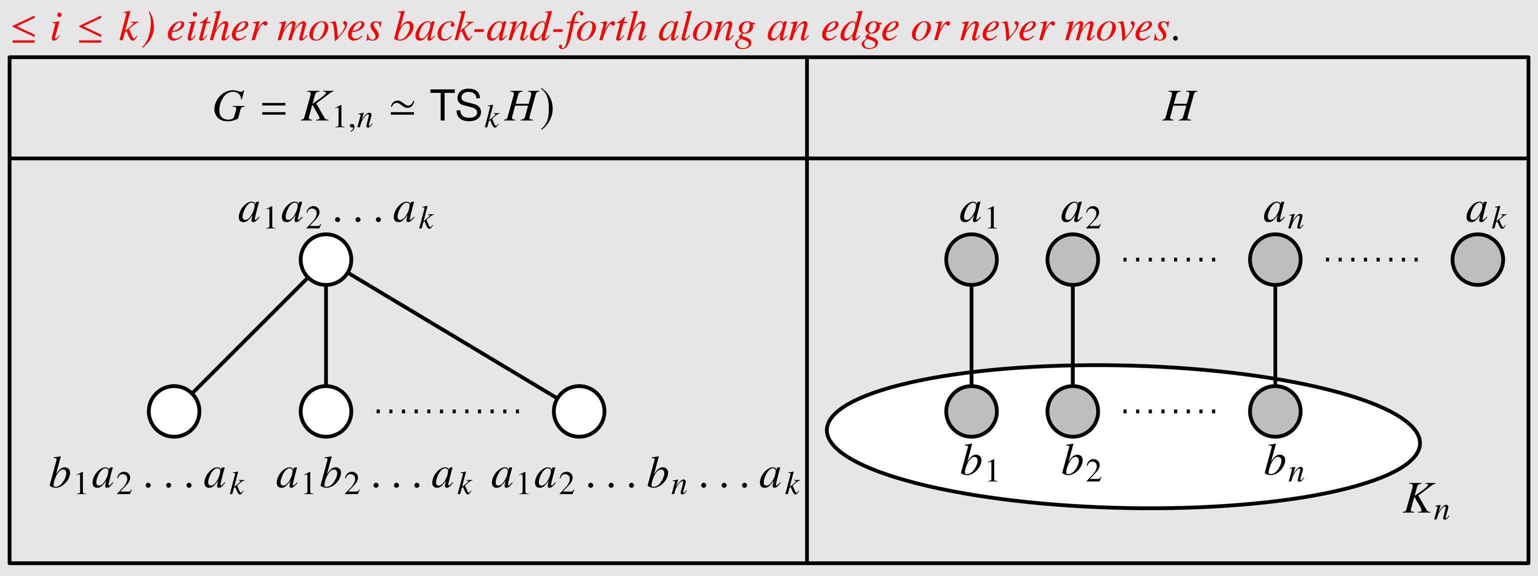

Figure 9. Design H s.t. $\mathsf{TS}_k(H) \simeq G$ ($k \geq 2$) where $G = K_{1,n}$ ($n \leq k$).

- **Open Problem:** G is a *tree*?
  - If there exists H s.t. $\mathsf{TS}_2(H) \simeq G$ for a tree G, H does not contain any P_5 as an induced subgraph.

References

- Avis, D. and **D. A. Hoang** (2022). "On Reconfiguration Graph of Independent Sets under Token Sliding". In Preparation.
- **Hoang, D. A.**, A. Suzuki, and T. Yagita (2020). "Reconfiguring k-path vertex covers". In: *Proceedings of WALCOM 2020*. Ed. by M. S. Rahman, K. Sadakane, and W.-K. Sung. Vol. 12049. LNCS. Springer, pp. 133–145. DOI: 10.1007/978-3-030-39881-1_12.
- **Hoang, D. A.** and R. Uehara (2016). "Sliding Tokens on a Cactus". In: *Proceedings of ISAAC 2016*. Vol. 64. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 37:1–37:26. DOI: 10.4230/LIPIcs.ISAAC.2016.37.
- Demaine, E. D., M. L. Demaine, E. Fox-Epstein, **D. A. Hoang**, T. Ito, H. Ono, Y. Otachi, R. Uehara, and T. 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.
- Fox-Epstein, E., **D. A. Hoang**, Y. Otachi, and R. Uehara (2015). "Sliding Token on Bipartite Permutation Graphs". In: *Proceedings of ISAAC 2015*. Vol. 9472. LNCS. Springer, pp. 237–247. DOI: 10.1007/978-3-662-48971-0_21.