On Token Sliding Reconfiguration Graphs of Independent Sets

Duc A. Hoang

VNU University of Science, Hanoi, Vietnam
hoanganhduc@hus.edu.vn

Based on discussion and collaboration with
David Avis (Kyoto University, Japan)
Meike Hatzel (IBS, Daejeon, South Korea)
Marcelo Garlet Milani (NII, Tokyo, Japan)

Workshop on Graphs and Beyond
(Hanoi, Vietnam)

March 24–27, 2026

Outline

TS$_k$-Graphs: Background and Motivation

TS_k-Graphs: Background and Motivation

For a graph G and fixed $k \geq 1$, $\mathsf{TS}_k(G)$ has:

- **vertices:** size-k independent sets of G (token-sets);
- **edges:** two sets I, J are adjacent iff $J = (I \setminus \{u\}) \cup \{v\}$ for some edge $uv \in E(G)$ and $v \notin I$ (i.e., sliding a token from u to an unoccupied neighbor v).

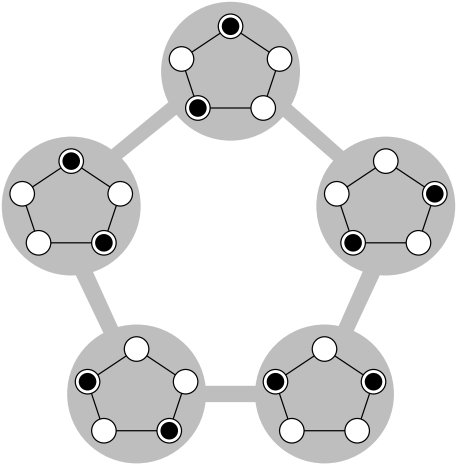


Figure: $\mathsf{TS}_2(C_5)$

TS_k-Graphs: Background and Motivation

General Framework: *Reconfiguration Problems*

- A *growing research area* in computer science which primarily aims to *understand the solution space of a computational problem*
- Given a computational problem Π and two feasible solutions S, T of Π, can we transform S into T by applying a sequence of *small changes* such that all intermediate states are also feasible solutions of Π?
- Under the "reconfiguration setting", $\mathsf{TS}_k(G)$ has been *extremely well-studied* from the *algorithmic perspective* [van den Heuvel 2013]; [Nishimura 2018]; [Bousquet, Mouawad, Nishimura, and Siebertz 2022]
- However, *the structural properties of TS_k-graphs* are still *largely unexplored*

TS_k-Graphs: Technical Challenges

- **Rapidly growing state space:** vertices are all size-k independent sets, so small changes in G may produce many new vertices and adjacencies.
- **Not hereditary:** being a TS_k-graph is not closed under induced subgraphs, even for trees. This prevents a simple forbidden-subgraph characterization as well as any inductive approach
- **Delicate dependence on k:** structural behaviour (e.g., appearance of cycles or realizability of families) can change sharply with small changes in k.
- **Inversion is hard:** building $\mathsf{TS}_k(G)$ from G is easy, but reconstructing a suitable G from a target graph requires careful constructions, labelings, and decompositions.

Paths Are TS_k-Graphs

Paths Are TS_k-Graphs

- We say that a graph H is a *TS_k-graph* if there exists a graph G such that $H \simeq \mathsf{TS}_k(G)$
  - For example, C_5 is a TS_2-graph, since $C_5 \simeq \mathsf{TS}_2(C_5)$
- Since $G \simeq \mathsf{TS}_1(G)$ for any graph G, *the case $k = 1$ is trivial and uninteresting*. From now on, we will always consider *fixed integers $k \geq 2$*
- In this section we illustrate some useful techniques by proving a simple result:

Theorem 1

A path P_n ($n \geq 1$) is a TS_k-graph for any $k \geq 2$

- We present three complementary approaches, each of which *indicates a way of looking at TS_k-graphs*
  1. Characterizing related graphs
  2. Explicit (brute-force) constructions
  3. Characterizing useful graph operations

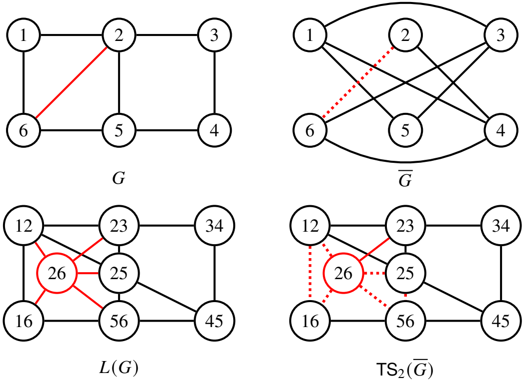
Line Graphs and TS_2-Graphs

Lemma 2

Let $\overline{G}$ and $L(G)$ be respectively the complement and line graph of G

(a) $\mathsf{TS}_2(\overline{G})$ *is a (spanning) subgraph of* $L(G)$

(b) $\mathsf{TS}_2(\overline{G}) \simeq L(G)$ *if and only if* G *is triangle-free*



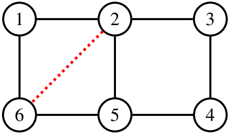
Line Graphs and TS_2-Graphs

Lemma 2

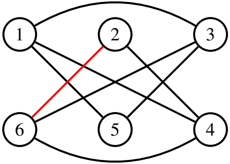
Let $\overline{G}$ and $L(G)$ be respectively the complement and line graph of G

(a) $\mathsf{TS}_2(\overline{G})$ *is a (spanning) subgraph of* $L(G)$

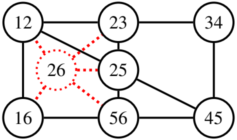
(b) $\mathsf{TS}_2(\overline{G}) \simeq L(G)$ *if and only if* G *is triangle-free*



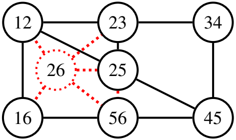
G



$\overline{G}$



$L(G)$



$\mathsf{TS}_2(\overline{G})$

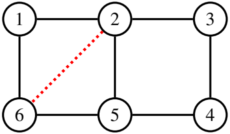
Line Graphs and TS_2-Graphs

Lemma 2

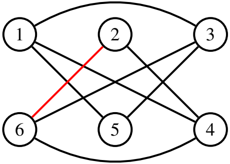
Let $\overline{G}$ and $L(G)$ be respectively the complement and line graph of G

(a) $\mathsf{TS}_2(\overline{G})$ *is a (spanning) subgraph of* $L(G)$

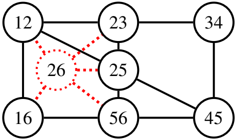
(b) $\mathsf{TS}_2(\overline{G}) \simeq L(G)$ *if and only if* G *is triangle-free*



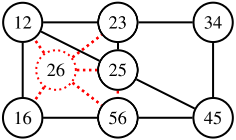
G



$\overline{G}$



$L(G)$



$\mathsf{TS}_2(\overline{G})$

- P_n is triangle-free
- $P_n \simeq L(P_{n+1})$
- Thus, $P_n \simeq \mathsf{TS}_2(\overline{P_{n+1}}) \Rightarrow$ P_n is a TS_2-graph

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

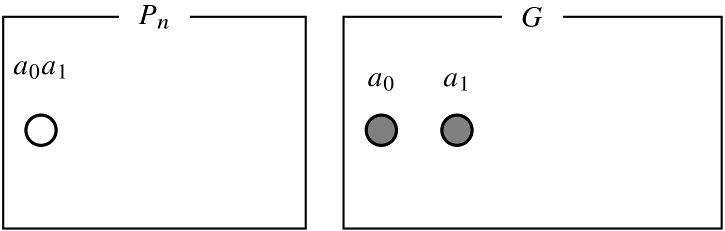


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

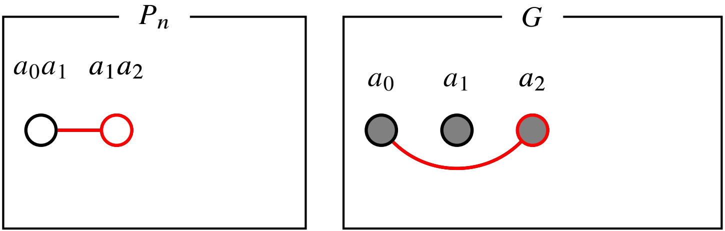


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

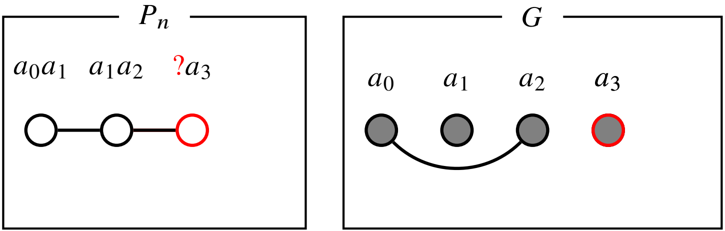


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

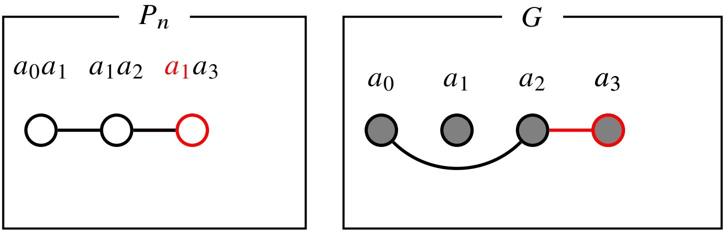


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

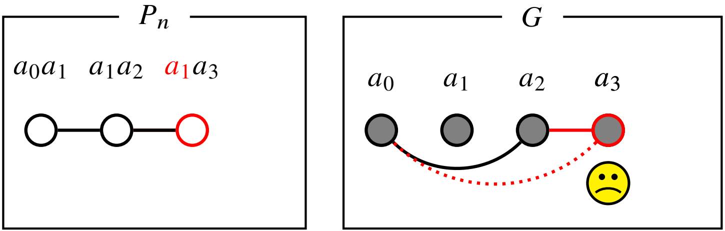


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

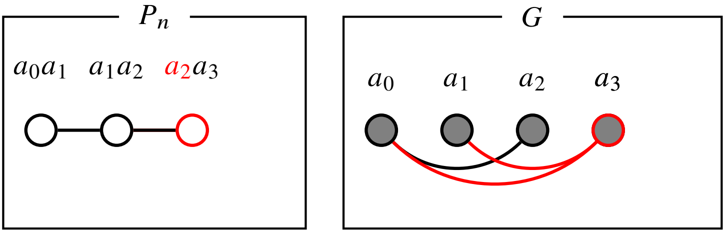


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

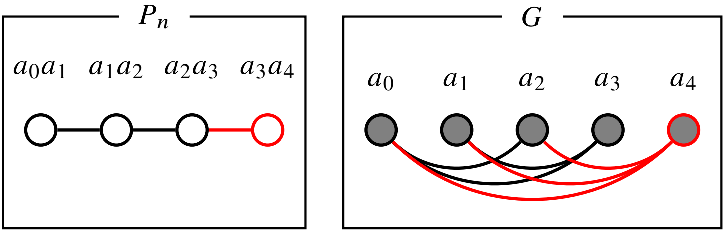


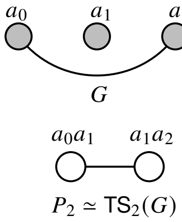
Figure: $P_n \simeq \mathsf{TS}_2(G)$

Gluing Labelled Graphs Together

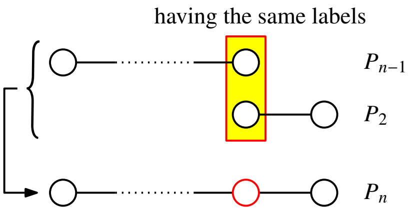
Key Idea ($k = 2$)

(a) There exists a (labelled) graph G such that $P_2 \simeq \mathsf{TS}_2(G)$

(b) With a "good" vertex-labelling (by stable sets of some graph), the TS_2-graph P_n can be constructed by *"gluing" two smaller TS_2-graphs P_{n-1} and P_2*, for any $n \geq 3$



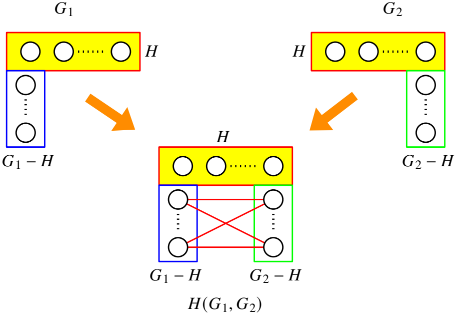
(a)



(b)

Gluing Labelled Graphs Together

- Vertex-labelled graphs G_1 and G_2 are *H-consistent* if *the (possibly empty) intersection of their vertex sets defines the same (possibly empty) common induced subgraph H*
- The *H-join of H-consistent graphs G_1, G_2*, denoted by $H(G_1, G_2)$, is the graph constructed by
  - Putting G_2 on top of G_1 (or vice versa) so that their common induced subgraph H coincide
  - Joining any pair of vertices v, w between $G_1 - H$ and $G_2 - H$, respectively



Gluing Labelled Graphs Together

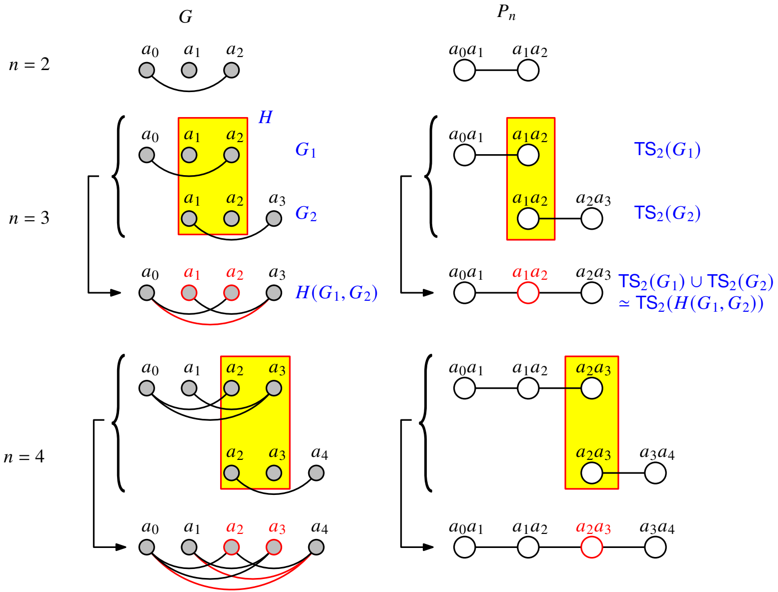


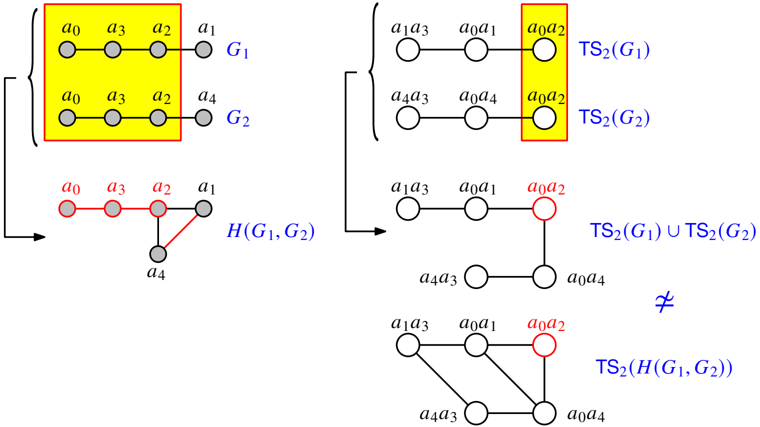
Figure: $P_n \simeq \mathsf{TS}_2(G)$

Gluing Labelled Graphs Together

Remark

There exist H-consistent graphs G_1, G_2 such that

$$\mathsf{TS}_2(G_1) \cup \mathsf{TS}_2(G_2) \not\simeq \mathsf{TS}_2(H(G_1, G_2))$$



Gluing Labelled Graphs Together

In general, the following proposition describes *how to compute the* TS_k*-graph of an* H*-join*

Proposition 3

Let $k \geq 2$ *and let* G_1 *and* G_2 *be two* H*-consistent graphs.* $\mathsf{TS}_k(H(G_1, G_2))$ *is the union of* $\mathsf{TS}_k(G_1)$, $\mathsf{TS}_k(G_2)$ *and for every pair of* k*-element stable sets* S_1 *in* G_1 *and* S_2 *in* G_2 *satisfying*

$$|S_1 \cap V(H)| = |S_2 \cap V(H)| = |S_1 \cap S_2| = k - 1, \tag{1}$$

the edge between S_1 *and* S_2

Gluing Labelled Graphs Together

In general, the following proposition describes *how to compute the* TS_k*-graph of an* H*-join*

Proposition 3

Let $k \geq 2$ *and let* G_1 *and* G_2 *be two* H*-consistent graphs.* $\mathsf{TS}_k(H(G_1, G_2))$ *is the union of* $\mathsf{TS}_k(G_1)$, $\mathsf{TS}_k(G_2)$ *and for every pair of* k*-element stable sets* S_1 *in* G_1 *and* S_2 *in* G_2 *satisfying*

$$|S_1 \cap V(H)| = |S_2 \cap V(H)| = |S_1 \cap S_2| = k - 1, \tag{1}$$

the edge between S_1 *and* S_2

- We say that $H(G_1, G_2)$ is *k-crossing free* if there are *no k-element stable sets satisfying condition (1) of Proposition 3*

Corollary 4

Let $k \geq 2$ *and let* G_1 *and* G_2 *be two* H*-consistent graphs.* $H(G_1, G_2)$ *is* k*-crossing free if and only if*

$$\mathsf{TS}_k(H(G_1, G_2)) \simeq \mathsf{TS}_k(G_1) \cup \mathsf{TS}_k(G_2) \tag{2}$$

Final Piece: How to solve for $k \geq 3$?

The following observation *settles the case $k \geq 3$* (note that $\alpha(\overline{P_{n+1}}) = 2$)

- Let $G = \mathsf{TS}_{\alpha(H)}(H)$.
- For $k \geq \alpha(H)$ let H' be H plus $k - \alpha(H)$ isolated vertices.
- Every size-k independent set of H' is a maximum independent set of H together with the added isolates, so moves in $\mathsf{TS}_k(H')$ correspond exactly to moves in $\mathsf{TS}_{\alpha(H)}(H)$.
- Hence $\mathsf{TS}_k(H') \simeq \mathsf{TS}_{\alpha(H)}(H)$.

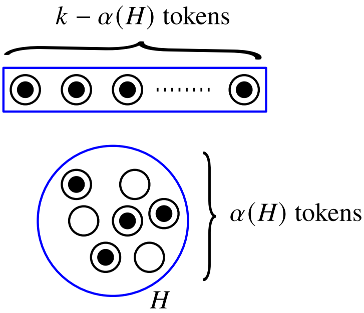


Figure: The graph H'

On TS_k-Graphs

On TS_k-Graphs

- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$)

On TS_k-Graphs

- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$)
- $K_n, \overline{K_n}$ $(n \geq 1)$ is a TS_k-graph

On TS_k-Graphs

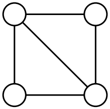
- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$)
- *K_n, $\overline{K_n}$ ($n \geq 1$)* is a TS_k-graph
- *P_n ($n \geq 1$) and C_n ($n \geq 3$)* are TS_k-graphs. On the other hand, *$\overline{P_n}$ ($n \geq 1$) and $\overline{C_n}$ ($n \geq 3$)* are TS_k-graphs for $n \leq 5$ and $n \leq 6$, respectively.

On TS_k-Graphs

- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$)
- K_n, $\overline{K_n}$ $(n \geq 1)$ is a TS_k-graph
- P_n $(n \geq 1)$ *and* C_n $(n \geq 3)$ are TS_k-graphs. On the other hand, $\overline{P_n}$ $(n \geq 1)$ *and* $\overline{C_n}$ $(n \geq 3)$ are TS_k-graphs for $n \leq 5$ and $n \leq 6$, respectively.
- $K_{m,n}$ $(1 \leq m \leq n)$ is a TS_k-graph for any $k \geq 2$ if and only if either $m = 1$ and $n \leq k$ or $m = n = 2$ (A similar approach can be used to characterize whether *split graphs* are (not) TS_k-graphs)

On TS_k-Graphs

- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$)
- *K_n, $\overline{K_n}$ ($n \geq 1$)* is a TS_k-graph
- *P_n ($n \geq 1$) and C_n ($n \geq 3$)* are TS_k-graphs. On the other hand, *$\overline{P_n}$ ($n \geq 1$) and $\overline{C_n}$ ($n \geq 3$)* are TS_k-graphs for $n \leq 5$ and $n \leq 6$, respectively.
- *$K_{m,n}$ ($1 \leq m \leq n$)* is a TS_k-graph for any $k \geq 2$ if and only if either $m = 1$ and $n \leq k$ or $m = n = 2$ (A similar approach can be used to characterize whether *split graphs* are (not) TS_k-graphs)
- For any $k \geq 2$,
  - *K_1* is the *smallest graph that is a TS_k-graph*
  - On the other hand, the *diamond* is the *smallest graph that is not a TS_k-graph*

G	H such that $G \simeq \mathsf{TS}_k(H)$ ($k \geq 2$)
 K_1	 k isolated vertices
 diamond	does not exist

On Acyclic TS_k-Graphs

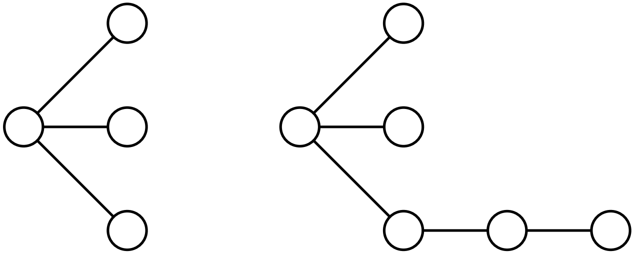
On Acyclic TS_k-Graphs

Remark

"Being a TS_k-graph" is *not* hereditary, even when $k = 2$

Using the "labelling vertices by stable sets" approach, we showed that

- $K_{1,3}$ is *not* a TS_2-graph. More generally, *$K_{1,n}$ is a TS_k-graph for some fixed $k \geq 2$ if and only if $n \leq k$*
- Replacing an edge of $K_{1,3}$ by a P_4 results a TS_2-graph



not a TS_2-graph   a TS_2-graph

On Acyclic TS_k-Graphs

Open Question

(1) Under which conditions a given graph G satisfies $\mathsf{TS}_k(G)$ is acyclic?

(2) Under which conditions trees are (not) TS_k-graphs?

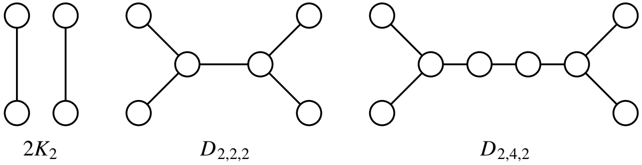
On Acyclic TS_k-Graphs

Open Question

(1) Under which conditions a given graph G satisfies $\mathsf{TS}_k(G)$ is acyclic?

(2) Under which conditions trees are (not) TS_k-graphs?

- Given a forest F
  - $\mathsf{TS}_2(F)$ is acyclic if and only if F is $\{2K_2, D_{2,2,2}\}$-free



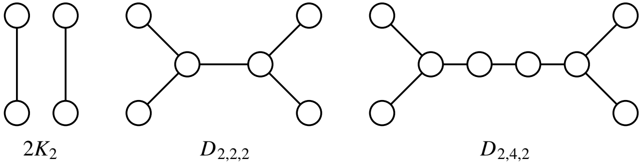
On Acyclic TS_k-Graphs

Open Question

(1) Under which conditions a given graph G satisfies $\mathsf{TS}_k(G)$ is acyclic?

(2) Under which conditions trees are (not) TS_k-graphs?

- Given a forest F
  - $\mathsf{TS}_2(F)$ is acyclic if and only if F is $\{2K_2, D_{2,2,2}\}$-free
  - $\mathsf{TS}_3(F)$ is acyclic if and only if F is $\{2K_2 + K_1, D_{2,2,2} + K_1, D_{2,4,2}\}$-free



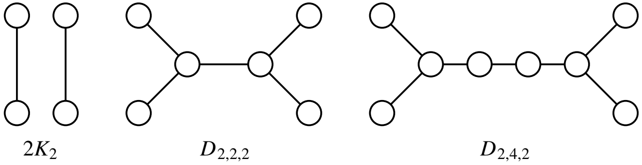
On Acyclic TS_k-Graphs

Open Question

(1) Under which conditions a given graph G satisfies $\mathsf{TS}_k(G)$ is acyclic?

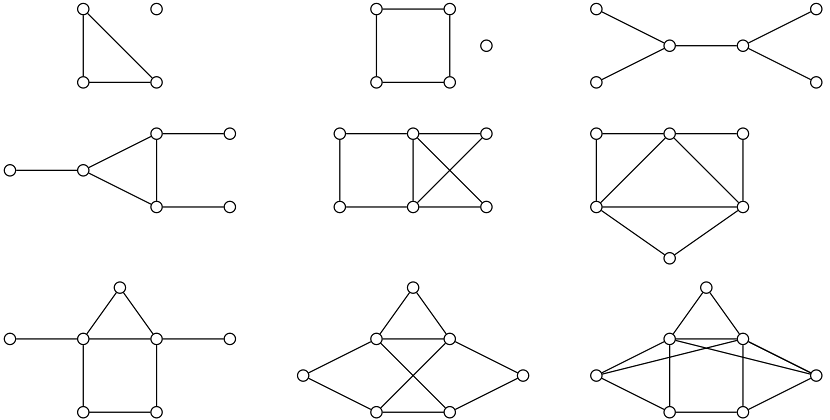
(2) Under which conditions trees are (not) TS_k-graphs?

- Given a forest F
  - $\mathsf{TS}_2(F)$ is acyclic if and only if F is $\{2K_2, D_{2,2,2}\}$-free
  - $\mathsf{TS}_3(F)$ is acyclic if and only if F is $\{2K_2 + K_1, D_{2,2,2} + K_1, D_{2,4,2}\}$-free
  - **Conjecture:** $\mathsf{TS}_k(F)$ is acyclic if and only if F is $\{2K_2 + (k-2)K_1, D_{2,2,2} + (k-2)K_1, D_{2,4,2} + (k-3)K_1\}$-free, for $k \geq 4$



On Acyclic TS_k-Graphs

- Given a graph G. If G contains either $\overline{C_n}$ *($n \geq 4$)* or *one of the following nine graphs* as an induced subgraph then $\mathsf{TS}_2(G)$ has a cycle
  - The $\overline{C_n}$ ($n \geq 4$) graphs come from the "line graph" approach (Lemma 2)
  - (Most of) The below nine graphs come from a computer program



- **Open Question:** Does the reverse hold? (i.e., Did we miss any graph?)

Thank you for your attention!

References

Bousquet, N., A. E. Mouawad, N. Nishimura, and S. Siebertz (2022). “A survey on the parameterized complexity of the independent set and (connected) dominating set reconfiguration problems”. In: *arXiv preprint*. arXiv: 2204.10526.

Nishimura, N. (2018). “Introduction to reconfiguration”. In: *Algorithms* 11.4. (article 52). DOI: 10.3390/a11040052.

van den Heuvel, J. (2013). “The complexity of change”. In: *Surveys in Combinatorics*. Vol. 409. London Math. Soc. Lecture Note Ser. Cambridge University Press, pp. 127–160. DOI: 10.1017/CBO9781139506748.005.