Token Jumping on Surfaces

Joint work with: Daniel W. Cranston and Moritz Mühlenthaler

Benjamin Peyrille

Université Grenoble Alpes

WG 2025, Otzenhausen, June 11 2025

Token sliding and jumping

Setting $G = (V, E)$: simple graph
I, J : independent sets of G of the same size $k \geq 1$
(we draw I as tokens ○ and J as tokens ⊕ placed on vertices of G)

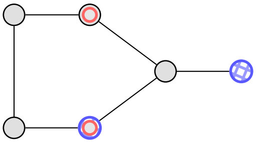
Question Can we transform I into J by moving tokens one-by-one while preserving the independent set property ?

Token sliding and jumping

Setting $G = (V, E)$: simple graph
I, J : independent sets of G of the same size $k \geq 1$
(we draw I as tokens ○ and J as tokens ⊛ placed on vertices of G)

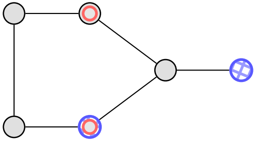
Question Can we transform I into J by moving tokens one-by-one while preserving the independent set property ?

TOKEN SLIDING



Slide along edges

TOKEN JUMPING



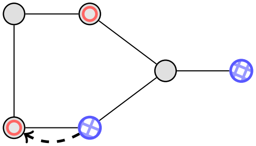
Jump anywhere

Token sliding and jumping

Setting $G = (V, E)$: simple graph
I, J : independent sets of G of the same size $k \geq 1$
(we draw I as tokens and J as tokens placed on vertices of G)

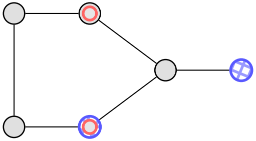
Question Can we transform I into J by moving tokens one-by-one while preserving the independent set property ?

<u>TOKEN SLIDING</u>



Slide along edges

<u>TOKEN JUMPING</u>



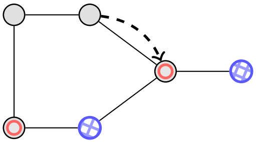
Jump anywhere

Token sliding and jumping

Setting $G = (V, E)$: simple graph
I, J : independent sets of G of the same size $k \geq 1$
(we draw I as tokens ○ and J as tokens ⊕ placed on vertices of G)

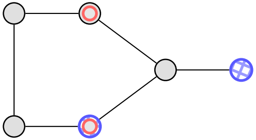
Question Can we transform I into J by moving tokens one-by-one while preserving the independent set property ?

<u>TOKEN SLIDING</u>



Slide along edges

<u>TOKEN JUMPING</u>



Jump anywhere

Token sliding and jumping

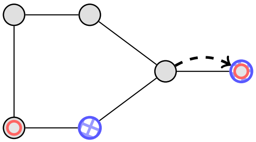
Setting $G = (V, E)$: simple graph

I, J : independent sets of G of the same size $k \geq 1$

(we draw I as tokens ○ and J as tokens ⊗ placed on vertices of G)

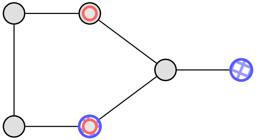
Question Can we transform I into J by moving tokens one-by-one while preserving the independent set property ?

TOKEN SLIDING



Slide along edges

TOKEN JUMPING



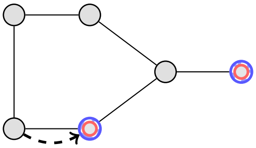
Jump anywhere

Token sliding and jumping

Setting $G = (V, E)$: simple graph
I, J : independent sets of G of the same size $k \geq 1$
(we draw I as tokens ○ and J as tokens ⊗ placed on vertices of G)

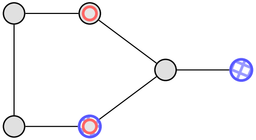
Question Can we transform I into J by moving tokens one-by-one while preserving the independent set property ?

TOKEN SLIDING



Slide along edges

TOKEN JUMPING



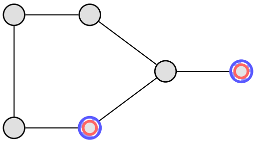
Jump anywhere

Token sliding and jumping

Setting $G = (V, E)$: simple graph
I, J : independent sets of G of the same size $k \geq 1$
(we draw I as tokens ○ and J as tokens ◎ placed on vertices of G)

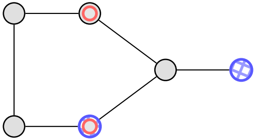
Question Can we transform I into J by moving tokens one-by-one while preserving the independent set property ?

<u>Token Sliding</u>



Slide along edges

<u>Token Jumping</u>



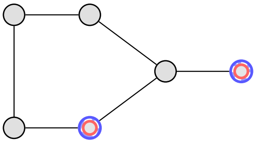
Jump anywhere

Token sliding and jumping

Setting $G = (V, E)$: simple graph
I, J : independent sets of G of the same size $k \geq 1$
(we draw I as tokens ○ and J as tokens ○ placed on vertices of G)

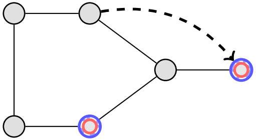
Question Can we transform I into J by moving tokens one-by-one while preserving the independent set property ?

<u>Token Sliding</u>



Slide along edges

<u>Token Jumping</u>



Jump anywhere

Complexity of Token Jumping

TOKEN JUMPING is

- PSPACE-complete for
  - subcubic planar graphs of bounded bandwidth (van der Zanden'14,Ito et al.'11,Hearn & Demaine'05)
  - perfect graphs (Kamiński et al., 2012)
- NP-complete for bipartite graphs (Lokshtanov & Mouawad, 2018)

Complexity of Token Jumping

Token Jumping is

- PSPACE-complete for
  - subcubic planar graphs of bounded bandwidth (van der Zanden'14,Ito et al.'11,Hearn & Demaine'05)
  - perfect graphs (Kamiński et al., 2012)
- NP-complete for bipartite graphs (Lokshtanov & Mouawad, 2018)

Token Jumping admits a polynomial-time algorithm for

- even-hole-free graphs (Kamiński et al., 2012)
- P_4-free graphs (Bonsma 2016, Bousquet & Bonamy 2012)
- claw-free graphs (Bonsma et al.'14)

Parameterized complexity of Token Jumping

A problem is **fixed-parameter tractable** (FPT) for some **parameter** k, if it admits an $O(f(k) \cdot \mathrm{poly}(n))$-time algorithm, where $f : \mathbb{N} \to \mathbb{N}$ is a computable function and n is the size of the instance.

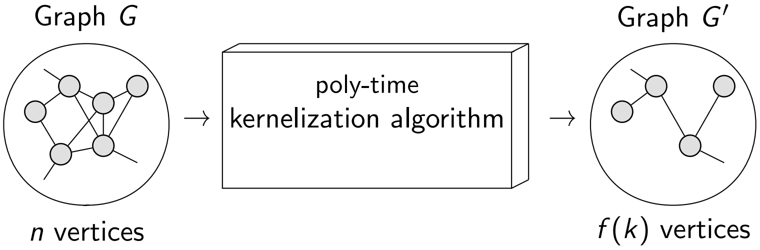
Parameterized complexity of Token Jumping

A problem is **fixed-parameter tractable** (FPT) for some **parameter** k, if it admits an $O(f(k) \cdot \text{poly}(n))$-time algorithm, where $f : \mathbb{N} \to \mathbb{N}$ is a computable function and n is the size of the instance.

Parameterized hardness (Mouawad, 2017)

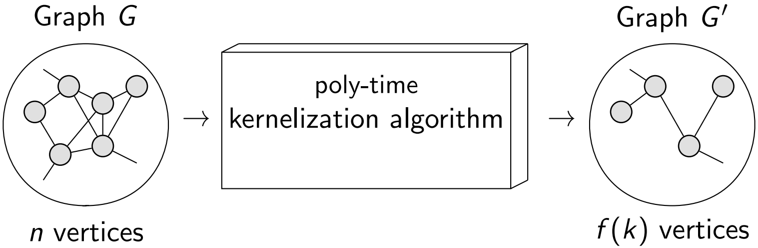
Token Jumping is $W[1]$-hard (<u>not FPT</u>) when parameterized by the number k of tokens.

Parameretized complexity: positive results



Kernelization $\implies$ FPT (bruteforce on $f(k)$ vertices)
If the function f is polynomial, the problem admits a **polynomial kernel**.

Parameretized complexity: positive results



Kernelization $\implies$ FPT (bruteforce on $f(k)$ vertices)
If the function f is polynomial, the problem admits a **polynomial kernel**.

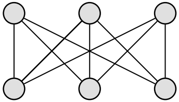
- FPT on planar graphs and $K_{3,t}$-free graphs (Ito et al., 2014)
- Polynomial kernel for $K_{t,t}$-free graphs (Bousquet et al., 2017)
- Polynomial kernel on graphs of bounded degeneracy (Lokshtanov et al., 2018)

Surfaces

The **genus** of a graph G is the smallest integer g such that G admits a crossing-free drawing on an orientable surface of genus g.

Surfaces

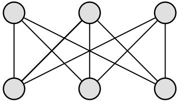
The **genus** of a graph G is the smallest integer g such that G admits a crossing-free drawing on an orientable surface of genus g.



$K_{3,3}$ is not planar ($g \neq 0$)

Surfaces

The **genus** of a graph G is the smallest integer g such that G admits a crossing-free drawing on an orientable surface of genus g.



$K_{3,3}$ is not planar ($g \neq 0$)

$K_{3,3}$ embedded on the torus ($g = 1$)

In a nutshell, the genus of a graph G is the smallest number of handles required to draw G on a mug.

Surfaces

The **genus** of a graph G is the smallest integer g such that G admits a crossing-free drawing on an orientable surface of genus g.

Main result (Cranston, Mühlenthaler, P., 2024)

Token Jumping parameterized by the genus g of the input graph and the number of tokens k admits a kernel of size $O((g+k)^2)$. Furthermore, the kernelization algorithm does not need to compute the genus.

Surfaces

The **genus** of a graph G is the smallest integer g such that G admits a crossing-free drawing on an orientable surface of genus g.

Main result (Cranston, Mühlenthaler, P., 2024)

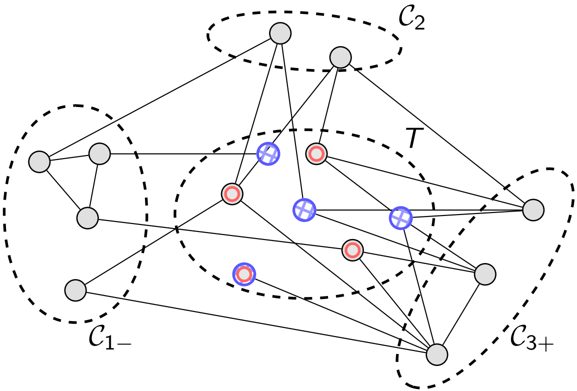
Token Jumping parameterized by the genus g of the input graph and the number of tokens k admits a kernel of size $O((g+k)^2)$. Furthermore, the kernelization algorithm does not need to compute the genus.

Kernelization results applied to graphs on surfaces:

Classes of graphs	Kernel size	For genus g
$K_{3,t}$-free (Ito et al., 14)	Ramsey$((2t+1)k, t+3)$	Ramsey$((8g+7)k, 4g+6)$
$K_{t,t}$-free (Bousquet et al., 17)	$O(f(t) \cdot k^{t \cdot 3^t})$	$O(h(g) \cdot k^{(4g+3) \cdot 3^{4g+3}})$
d-degenerate (Lokshtanov et al., 18)	$(2d+1)(2d+1)!(2k-1)^{2d+1}$	$(2H(g)-1)(2H(g)-1)!(2k-1)^{2H(g)-1}$
all graphs (This presentation!)	$O((g+k)^2)$	-

First step: Partition

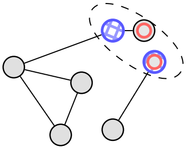
- T: vertices with a token ($T = I \cup J$)
- $\mathcal{C}_{1-}$: vertices adjacent to at most one element of T
- $\mathcal{C}_2$: vertices adjacent to exactly two elements of T
- $\mathcal{C}_{3+}$: vertices adjacent to least three elements of T



$\mathcal{C}_{1-}$ and $\mathcal{C}_{3+}$: easily bounded

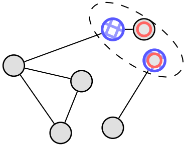
The Heawood number $H(g) = \left\lfloor (7 + \sqrt{1 + 48g})/2 \right\rfloor$ is the largest number of colors required to properly color any graph of genus g. If $|\mathcal{C}_{1-}| \geq H(g) \cdot k$, the instance is $\textsc{Yes}$. So we can assume

$$|\mathcal{C}_{1-}| < H(g) \cdot k = O(\sqrt{g} \cdot k).$$



$\mathcal{C}_{1-}$ and $\mathcal{C}_{3+}$: easily bounded

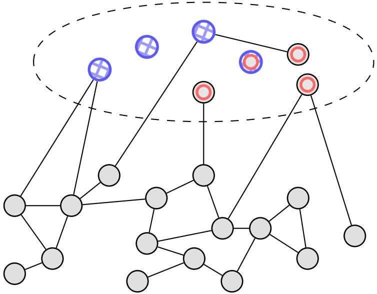
The Heawood number $H(g) = \left\lfloor (7 + \sqrt{1 + 48g})/2 \right\rfloor$ is the largest number of colors required to properly color any graph of genus g.
If $|\mathcal{C}_{1-}| \geq H(g) \cdot k$, the instance is YES. So we can assume



$$|\mathcal{C}_{1-}| < H(g) \cdot k = O(\sqrt{g} \cdot k).$$

$k = 4$

$|\mathcal{C}_{1-}| = 4 \cdot 4 = 16$

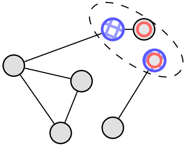


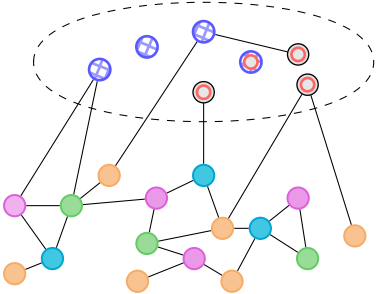
Planar: $H(g) = 4$

$\mathcal{C}_{1-}$ and $\mathcal{C}_{3+}$: easily bounded

The Heawood number $H(g) = \left\lfloor (7 + \sqrt{1 + 48g})/2 \right\rfloor$ is the largest number of colors required to properly color any graph of genus g.
If $|\mathcal{C}_{1-}| \geq H(g) \cdot k$, the instance is YES. So we can assume

$$|\mathcal{C}_{1-}| < H(g) \cdot k = O(\sqrt{g} \cdot k).$$





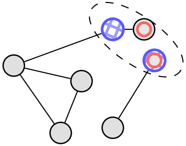
$k = 4$

$|\mathcal{C}_{1-}| = 4 \cdot 4 = 16$

Planar: $H(g) = 4$

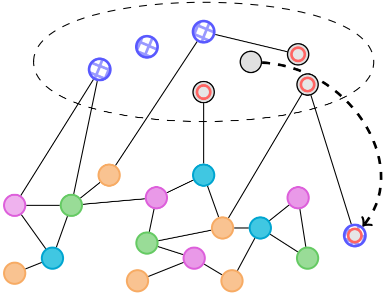
$\mathcal{C}_{1-}$ and $\mathcal{C}_{3+}$: easily bounded

The Heawood number $H(g) = \left\lfloor (7 + \sqrt{1 + 48g})/2 \right\rfloor$ is the largest number of colors required to properly color any graph of genus g.
If $|\mathcal{C}_{1-}| \geq H(g) \cdot k$, the instance is $\textsc{Yes}$. So we can assume



$$|\mathcal{C}_{1-}| < H(g) \cdot k = O(\sqrt{g} \cdot k).$$

$|\mathcal{C}_{1-}| = 4 \cdot 4 = 16$

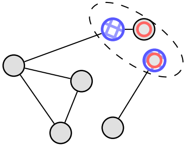


$k = 4$

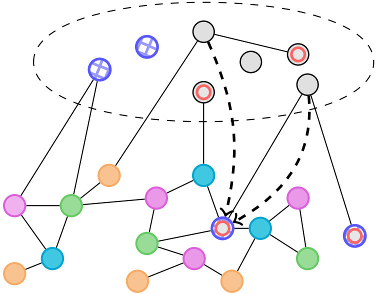
Planar: $H(g) = 4$

$\mathcal{C}_{1-}$ and $\mathcal{C}_{3+}$: easily bounded

The Heawood number $H(g) = \left\lfloor (7 + \sqrt{1 + 48g})/2 \right\rfloor$ is the largest number of colors required to properly color any graph of genus g.
If $|\mathcal{C}_{1-}| \geq H(g) \cdot k$, the instance is $\textsc{Yes}$. So we can assume



$$|\mathcal{C}_{1-}| < H(g) \cdot k = O(\sqrt{g} \cdot k).$$



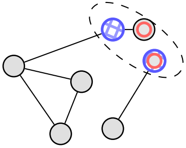
$k = 4$

$|\mathcal{C}_{1-}| = 4 \cdot 4 = 16$

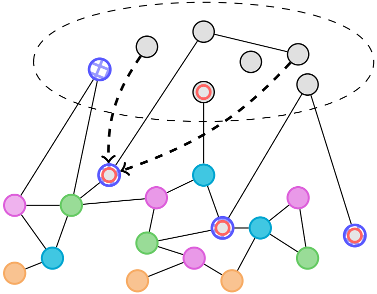
Planar: $H(g) = 4$

$\mathcal{C}_{1-}$ and $\mathcal{C}_{3+}$: easily bounded

The Heawood number $H(g) = \left\lfloor (7 + \sqrt{1 + 48g})/2 \right\rfloor$ is the largest number of colors required to properly color any graph of genus g.
If $|\mathcal{C}_{1-}| \geq H(g) \cdot k$, the instance is YES. So we can assume



$$|\mathcal{C}_{1-}| < H(g) \cdot k = O(\sqrt{g} \cdot k).$$



$k = 4$

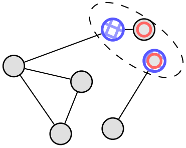
$|\mathcal{C}_{1-}| = 4 \cdot 4 = 16$

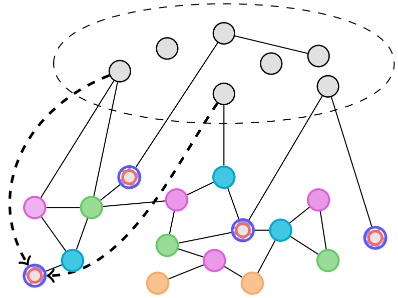
Planar: $H(g) = 4$

$\mathcal{C}_{1-}$ and $\mathcal{C}_{3+}$: easily bounded

The Heawood number $H(g) = \left\lfloor (7 + \sqrt{1 + 48g})/2 \right\rfloor$ is the largest number of colors required to properly color any graph of genus g.
If $|\mathcal{C}_{1-}| \geq H(g) \cdot k$, the instance is YES. So we can assume

$$|\mathcal{C}_{1-}| < H(g) \cdot k = O(\sqrt{g} \cdot k).$$





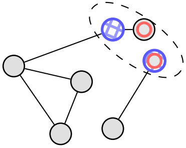
$k = 4$

$|\mathcal{C}_{1-}| = 4 \cdot 4 = 16$

Planar: $H(g) = 4$

$\mathcal{C}_{1-}$ and $\mathcal{C}_{3+}$: easily bounded

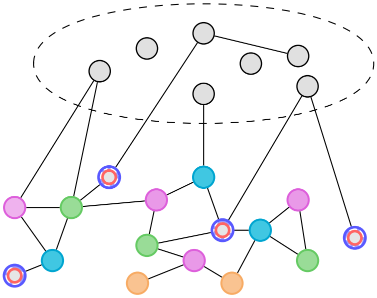
The Heawood number $H(g) = \left\lfloor (7 + \sqrt{1+48g})/2 \right\rfloor$ is the largest number of colors required to properly color any graph of genus g.
If $|\mathcal{C}_{1-}| \geq H(g) \cdot k$, the instance is $\textsc{Yes}$. So we can assume



$$|\mathcal{C}_{1-}| < H(g) \cdot k = O(\sqrt{g} \cdot k).$$

$k = 4$

$|\mathcal{C}_{1-}| = 4 \cdot 4 = 16$

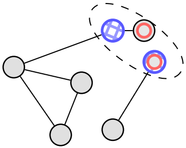


Planar: $H(g) = 4$

$\mathcal{C}_{1-}$ and $\mathcal{C}_{3+}$: easily bounded

The Heawood number $H(g) = \lfloor (7 + \sqrt{1 + 48g})/2 \rfloor$ is the largest number of colors required to properly color any graph of genus g.
If $|\mathcal{C}_{1-}| \geq H(g) \cdot k$, the instance is Yes. So we can assume

$$|\mathcal{C}_{1-}| < H(g) \cdot k = O(\sqrt{g} \cdot k).$$

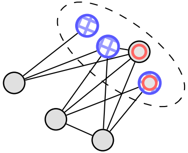


Theorem (Bouchet, 1978)

A graph of genus g cannot contain $K_{3,4g+3}$ as a subgraph.

Using an auxillary graph, we can use Euler's formula to get

$$|\mathcal{C}_{3+}| \leq 16g^2 + 16gk + 8k.$$



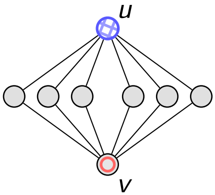
$\mathcal{C}_2$: not clear yet

Let $C_{\{u,v\}}$ be the **projection class** of $\{u, v\} \subseteq T$, that is $\{w : w \in V - T \text{ s.t } N_T(w) = \{u, v\}\}$.
Let $\{u, v\}$ such that $C_{\{u,v\}} \neq \emptyset$.

$\mathcal{C}_2$: not clear yet

Let $C_{\{u,v\}}$ be the **projection class** of $\{u, v\} \subseteq T$, that is $\{w : w \in V - T \text{ s.t } N_T(w) = \{u, v\}\}$.
Let $\{u, v\}$ such that $C_{\{u,v\}} \neq \emptyset$.
There can be an arbitrary number of vertices in $C_{\{u,v\}}$:

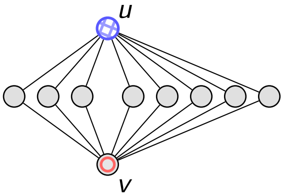
$C_{\{u,v\}}$



$\mathcal{C}_2$: not clear yet

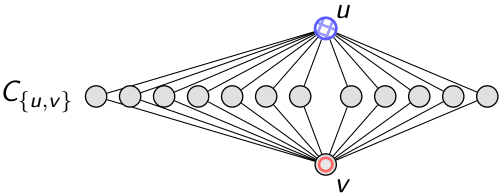
Let $C_{\{u,v\}}$ be the **projection class** of $\{u, v\} \subseteq T$, that is $\{w : w \in V - T \text{ s.t } N_T(w) = \{u, v\}\}$.
Let $\{u, v\}$ such that $C_{\{u,v\}} \neq \emptyset$.
There can be an arbitrary number of vertices in $C_{\{u,v\}}$:

$C_{\{u,v\}}$



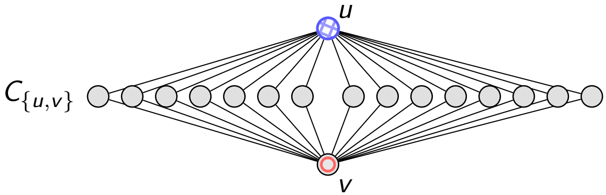
$\mathcal{C}_2$: not clear yet

Let $C_{\{u,v\}}$ be the **projection class** of $\{u, v\} \subseteq T$, that is $\{w : w \in V - T \text{ s.t } N_T(w) = \{u, v\}\}$.
Let $\{u, v\}$ such that $C_{\{u,v\}} \neq \emptyset$.
There can be an arbitrary number of vertices in $C_{\{u,v\}}$:



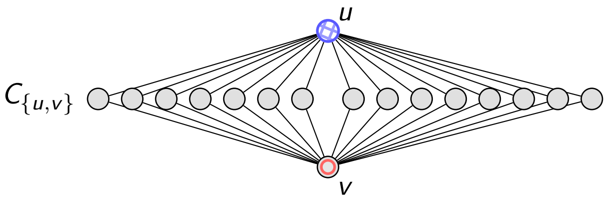
$\mathcal{C}_2$: not clear yet

Let $C_{\{u,v\}}$ be the **projection class** of $\{u, v\} \subseteq T$, that is $\{w : w \in V - T \text{ s.t } N_T(w) = \{u, v\}\}$.
Let $\{u, v\}$ such that $C_{\{u,v\}} \neq \emptyset$.
There can be an arbitrary number of vertices in $C_{\{u,v\}}$:



$\mathcal{C}_2$: not clear yet

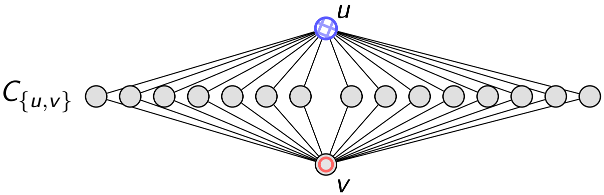
Let $C_{\{u,v\}}$ be the **projection class** of $\{u, v\} \subseteq T$, that is $\{w : w \in V - T \text{ s.t } N_T(w) = \{u, v\}\}$.
Let $\{u, v\}$ such that $C_{\{u,v\}} \neq \emptyset$.
There can be an arbitrary number of vertices in $C_{\{u,v\}}$:



Our goal: reduce $|\mathcal{C}_2|$ to size $O((g + k)^2)$.

$\mathcal{C}_2$: not clear yet

Let $C_{\{u,v\}}$ be the **projection class** of $\{u, v\} \subseteq T$, that is $\{w : w \in V - T \text{ s.t } N_T(w) = \{u, v\}\}$.
Let $\{u, v\}$ such that $C_{\{u,v\}} \neq \emptyset$.
There can be an arbitrary number of vertices in $C_{\{u,v\}}$:



Our goal: reduce $|\mathcal{C}_2|$ to size $O((g + k)^2)$.

By Euler's formula, the number of non-empty projection classes is at most $6k + 6g$.

If $|C_{\{u,v\}}| > 8g + 4k$, we can replace $C_{\{u,v\}}$ by an independent set of size $2k$.

Planar zones

Theorem (Malnič and Mohar, 1992)

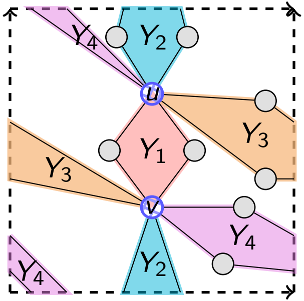
The maximum number of non-homotopic internally disjoint u, v-paths on any graph of genus g is $\max(1, 4g)$.

Planar zones

Theorem (Malnič and Mohar, 1992)

The maximum number of non-homotopic internally disjoint u, v-paths on any graph of genus g is $\max(1, 4g)$.

Hence, paths between u and v in $C_{\{u,v\}}$ divide the surface in at most $4g$ **planar zones**.

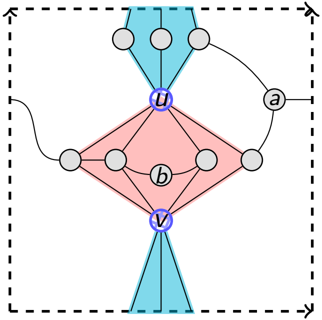


Four zones for $C_{\{u,v\}}$ on a torus.

Anatomy of the zone

Each zone has two **outer** vertices and some **inner** vertices.

Inner vertices form induced linear forests in $C_{\{u,v\}}$ whose independent sets are large and easy to find.



- Vertices outside a zone cannot be adjacent to inner vertices of $C_{\{u,v\}}$.
- Vertices inside a zone can only be adjacent to two vertices of $C_{\{u,v\}}$.

Problem solved

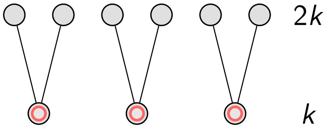
$$
\begin{aligned}
C_{\{u,v\}} \text{ is large } (8g+4k) \quad &\Longrightarrow \quad \geq 4k \quad \text{inner vertices} \\
&\Longrightarrow \quad \geq 4k \quad \text{size linear forest} \\
&\Longrightarrow \quad 2k \quad \text{size independent set } T_{\{u,v\}} \text{ in } C_{\{u,v\}}
\end{aligned}
$$

Recall each token of I is adjacent to at most two inner vertices of $C_{\{u,v\}}$.
If $C_{u,v}$ is used in a reconfiguration sequence, we can move all tokens from I to $T_{\{u,v\}}$ and do the same for J.

Problem solved

$$
\begin{array}{llrl}
C_{\{u,v\}} \text{ is large } (8g+4k) & \implies & \geq 4k & \text{inner vertices} \\
& \implies & \geq 4k & \text{size linear forest} \\
& \implies & 2k & \text{size independent set } T_{\{u,v\}} \text{ in } C_{\{u,v\}}
\end{array}
$$

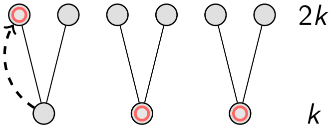
Recall each token of I is adjacent to at most two inner vertices of $C_{\{u,v\}}$.
If $C_{u,v}$ is used in a reconfiguration sequence, we can move all tokens from I to $T_{\{u,v\}}$ and do the same for J.



Problem solved

$$
\begin{aligned}
C_{\{u,v\}} \text{ is large } (8g+4k) \quad &\implies \quad \geq 4k \quad \text{inner vertices} \\
&\implies \quad \geq 4k \quad \text{size linear forest} \\
&\implies \quad 2k \quad \text{size independent set } T_{\{u,v\}} \text{ in } C_{\{u,v\}}
\end{aligned}
$$

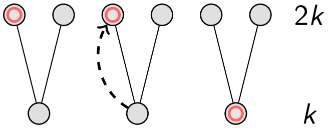
Recall each token of I is adjacent to at most two inner vertices of $C_{\{u,v\}}$.
If $C_{u,v}$ is used in a reconfiguration sequence, we can move all tokens from I to $T_{\{u,v\}}$ and do the same for J.



Problem solved

$$
\begin{aligned}
C_{\{u,v\}} \text{ is large } (8g+4k) \quad &\Longrightarrow \quad \geq 4k \quad \text{inner vertices} \\
&\Longrightarrow \quad \geq 4k \quad \text{size linear forest} \\
&\Longrightarrow \quad 2k \quad \text{size independent set } T_{\{u,v\}} \text{ in } C_{\{u,v\}}
\end{aligned}
$$

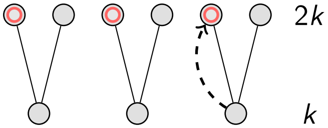
Recall each token of I is adjacent to at most two inner vertices of $C_{\{u,v\}}$.
If $C_{u,v}$ is used in a reconfiguration sequence, we can move all tokens from I to $T_{\{u,v\}}$ and do the same for J.



Problem solved

$$
\begin{array}{llll}
C_{\{u,v\}} \text{ is large } (8g+4k) & \Longrightarrow & \geq 4k & \text{inner vertices} \\
 & \Longrightarrow & \geq 4k & \text{size linear forest} \\
 & \Longrightarrow & 2k & \text{size independent set } T_{\{u,v\}} \text{ in } C_{\{u,v\}}
\end{array}
$$

Recall each token of I is adjacent to at most two inner vertices of $C_{\{u,v\}}$.
If $C_{u,v}$ is used in a reconfiguration sequence, we can move all tokens from I to $T_{\{u,v\}}$ and do the same for J.



Problem solved

$$\begin{aligned} C_{\{u,v\}} \text{ is large } (8g+4k) \quad &\implies \quad \geq 4k \quad \text{inner vertices} \\ &\implies \quad \geq 4k \quad \text{size linear forest} \\ &\implies \quad 2k \quad \text{size independent set } T_{\{u,v\}} \text{ in } C_{\{u,v\}} \end{aligned}$$

Recall each token of I is adjacent to at most two inner vertices of $C_{\{u,v\}}$.
If $C_{u,v}$ is used in a reconfiguration sequence, we can move all tokens from I to $T_{\{u,v\}}$ and do the same for J.

So we can assume all $C_{\{u,v\}}$ are of size at most $8g+4k$.

Problem solved... or is it?

$$\begin{aligned} C_{\{u,v\}} \text{ is large } (8g+4k) \quad &\Longrightarrow \quad \geq 4k \quad \text{inner vertices} \\ &\Longrightarrow \quad \geq 4k \quad \text{size linear forest} \\ &\Longrightarrow \quad 2k \quad \text{size independent set } T_{\{u,v\}} \text{ in } C_{\{u,v\}} \end{aligned}$$

Recall each token of I is adjacent to at most two inner vertices of $C_{\{u,v\}}$.
If $C_{u,v}$ is used in a reconfiguration sequence, we can move all tokens from I to $T_{\{u,v\}}$ and do the same for J.

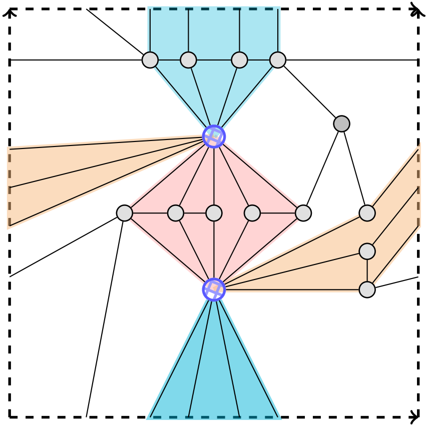
So we can assume all $C_{\{u,v\}}$ are of size at most $8g + 4k$.

Problem: knowing the genus of the graph or a crossing-free drawing, is hard.

We will find that large linear forest without any information on the genus.

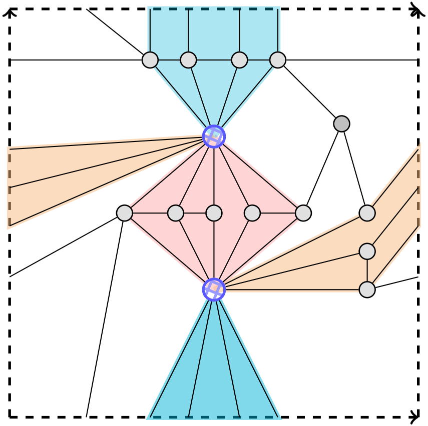
The algorithm

1 $Z := C_{\{u,v\}}$



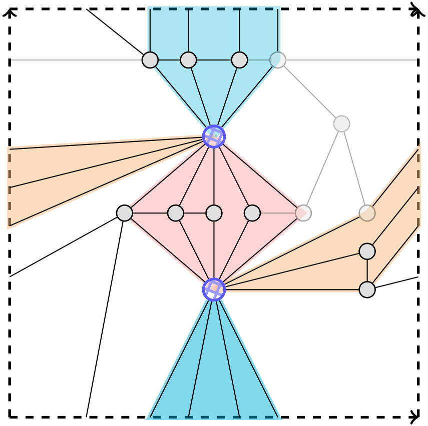
The algorithm

$Z := C_{\{u,v\}}$
for $v \in V - (C_{\{u,v\}} \cup Y)$ **do**
  if *v has at least 3 neighbors in $C_{\{u,v\}}$* **then**
    $Z \leftarrow Z - N(v)$



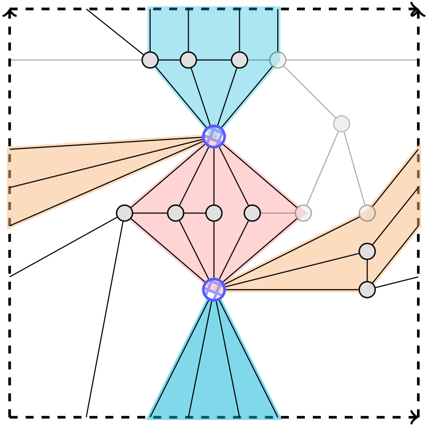
The algorithm

$Z := C_{\{u,v\}}$
for $v \in V - (C_{\{u,v\}} \cup Y)$ **do**
    if v *has at least* 3 *neighbors in* $C_{\{u,v\}}$ **then**
        $Z \leftarrow Z - N(v)$



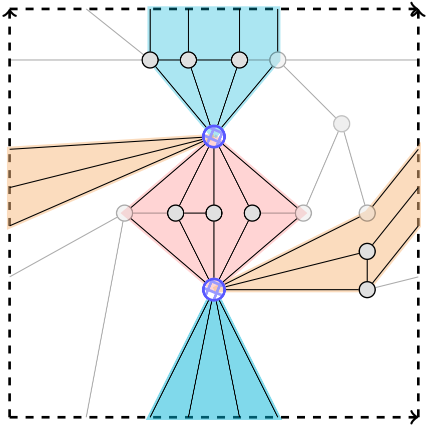
The algorithm

$Z := C_{\{u,v\}}$
for $v \in V - (C_{\{u,v\}} \cup Y)$ **do**
  if *v has at least 3 neighbors in $C_{\{u,v\}}$* **then**
    $Z \leftarrow Z - N(v)$
for $w \in Z$ **do**
  if *w has degree at least 3 in $G[Z]$* **then**
    $Z \leftarrow Z - w$



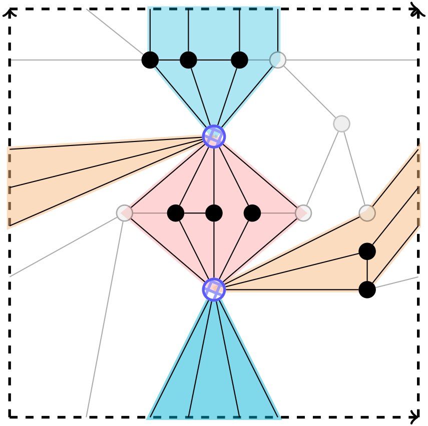
The algorithm

```
Z := C_{u,v}
for v ∈ V − (C_{u,v} ∪ Y) do
    if v has at least 3 neighbors in C_{u,v} then
        Z ← Z − N(v)
for w ∈ Z do
    if w has degree at least 3 in G[Z] then
        Z ← Z − w
```



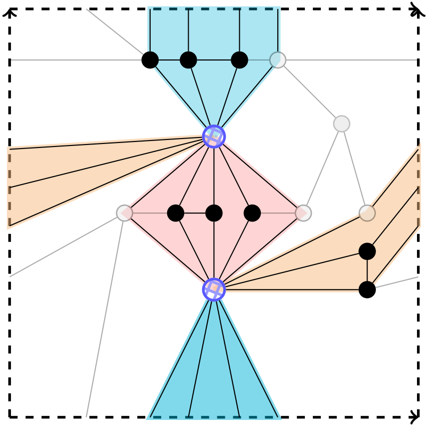
The algorithm

$Z := C_{\{u,v\}}$
for $v \in V - (C_{\{u,v\}} \cup Y)$ **do**
  if *v has at least 3 neighbors in $C_{\{u,v\}}$* **then**
    $Z \leftarrow Z - N(v)$
for $w \in Z$ **do**
  if *w has degree at least 3 in $G[Z]$* **then**
    $Z \leftarrow Z - w$
Remove arbitrarily one vertex from each cycle in $G[Z]$
return Z



The algorithm

$Z := C_{\{u,v\}}$
for $v \in V - (C_{\{u,v\}} \cup Y)$ **do**
  if *v has at least 3 neighbors in $C_{\{u,v\}}$* **then**
    $Z \leftarrow Z - N(v)$
for $w \in Z$ **do**
  if *w has degree at least 3 in $G[Z]$* **then**
    $Z \leftarrow Z - w$
Remove arbitrarily one vertex from each cycle in $G[Z]$
return Z



This procedure outputs a linear forest of size at least equal to the number of inner vertices, without any information on the genus.

Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Open question:

- Does Token Jumping admit a kernel of size $O(g^2 + gk + k)$?
- What other problems can be parameterized in such a way?

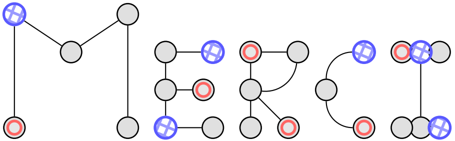
Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Open question:

- Does Token Jumping admit a kernel of size $O(g^2 + gk + k)$?
  - A recent result of Cranston and Bousquet seems to imply a kernel of size $O(g^{2.5} + g^{1.5}k + k)$.
- What other problems can be parameterized in such a way?



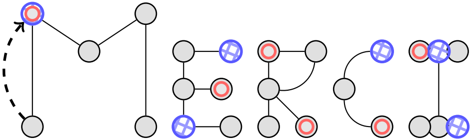
Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Open question:

- Does Token Jumping admit a kernel of size $O(g^2 + gk + k)$?
  - A recent result of Cranston and Bousquet seems to imply a kernel of size $O(g^{2.5} + g^{1.5}k + k)$.
- What other problems can be parameterized in such a way?



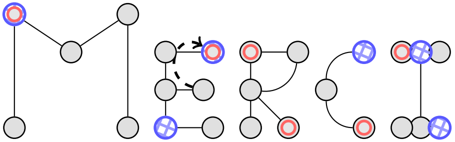
Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Open question:

- Does Token Jumping admit a kernel of size $O(g^2 + gk + k)$?
  - A recent result of Cranston and Bousquet seems to imply a kernel of size $O(g^{2.5} + g^{1.5}k + k)$.
- What other problems can be parameterized in such a way?



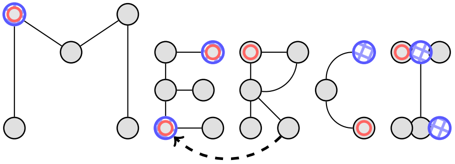
Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Open question:

- Does Token Jumping admit a kernel of size $O(g^2 + gk + k)$?
  - A recent result of Cranston and Bousquet seems to imply a kernel of size $O(g^{2.5} + g^{1.5}k + k)$.
- What other problems can be parameterized in such a way?



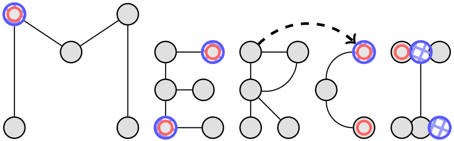
Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Open question:

- Does Token Jumping admit a kernel of size $O(g^2 + gk + k)$?
  - A recent result of Cranston and Bousquet seems to imply a kernel of size $O(g^{2.5} + g^{1.5}k + k)$.
- What other problems can be parameterized in such a way?



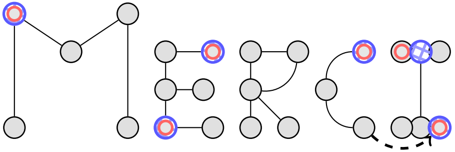
Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Open question:

- Does Token Jumping admit a kernel of size $O(g^2 + gk + k)$?
  - A recent result of Cranston and Bousquet seems to imply a kernel of size $O(g^{2.5} + g^{1.5}k + k)$.
- What other problems can be parameterized in such a way?



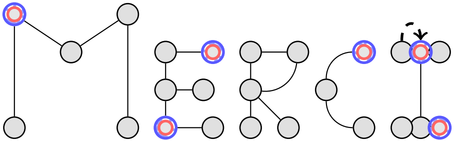
Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Open question:

- Does Token Jumping admit a kernel of size $O(g^2 + gk + k)$?
  - A recent result of Cranston and Bousquet seems to imply a kernel of size $O(g^{2.5} + g^{1.5}k + k)$.
- What other problems can be parameterized in such a way?



Conclusion

We obtain a kernel of size $O((g+k)^2)$ for Token Jumping.

Our algorithm is very simple and requires no information on the genus of the input graph.

Open question:

- Does Token Jumping admit a kernel of size $O(g^2 + gk + k)$?
  - A recent result of Cranston and Bousquet seems to imply a kernel of size $O(g^{2.5} + g^{1.5}k + k)$.
- What other problems can be parameterized in such a way?