The Computational Complexity of Positive Non-Clashing Teaching in Graphs

by Robert Ganian[1], Liana Khazaliya[1], Fionn Mc Inerney[2], and Mathis Rocton[1]

[1]TU Wien (Austria), [2]Telefónica Scientific Research (Spain)

Positive Non-Clashing Teaching

Strict Non-Clash

Input: Graph G and an integer k.

Question: Is there a positive non-clashing teaching map for the set of all balls of G with dimension at most k?

Non-Clash

Input: A graph G, a set $\mathcal{B}$ of balls of G, and an integer k.

Question: Is there a positive non-clashing teaching map for $\mathcal{B}$ with dimension at most k?

Related Work

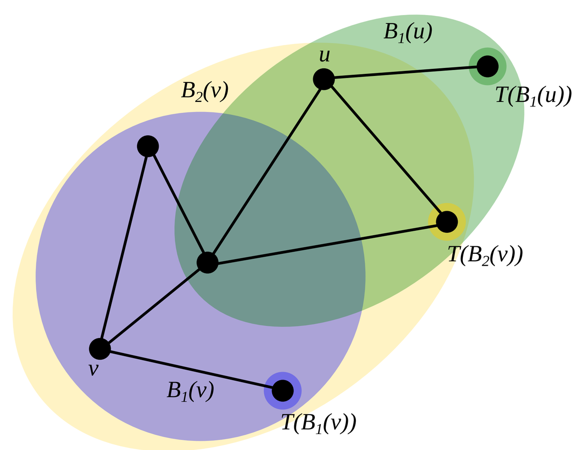
Introduced the concept
[Kirkpatrick et al., 2019; Fallat et al., 2023]

NP-hardness of Non-Clash
[Kirkpatrick et al., 2019]

Relation to recursive teaching
[Simon, 2023]

Relation to sample compression schemes
[Kirkpatrick et al., 2019; Fallat et al., 2023; Chalopin et al., 2023]

For Strict Non-Clash: NP-hardness for large dimension, running time upper and lower bounds, tight fixed-parameter algorithm when parameterized by the vertex cover number of G [Chalopin et al., 2024]



Notation and Definitions

Let G be a simple, finite, and undirected graph.

For an integer $r \geq 0$ and a vertex $v \in V(G)$, the *ball* $B_r(v)$ is the set of all vertices at distance at most r from its *center* v.

Let $\mathcal{B}$ be a set of balls of G.

A *positive teaching map* T for $\mathcal{B}$ is a mapping which assigns to each ball $B \in \mathcal{B}$ a *teaching set* $T(B) \subseteq B$, *i.e.*, a subset of the vertices of B.

The *dimension* of T is $\max_{B \in \mathcal{B}} |T(B)|$.

A positive teaching map T is *non-clashing* for $\mathcal{B}$ if, for each pair of distinct balls $B_1, B_2 \in \mathcal{B}$, there exists a vertex $w \in T(B_1) \cup T(B_2)$ such that $w \notin B_1 \cap B_2$.

The vertex w *distinguishes* B_1 and B_2, or distinguishes B_1 from B_2 (or vice versa).

If a teaching map is not non-clashing, there is a *conflict* between any two balls for which there is no element distinguishing them.

Main contributions:

(1) NP-hardness of Strict Non-Clash, even when $k = 2$.

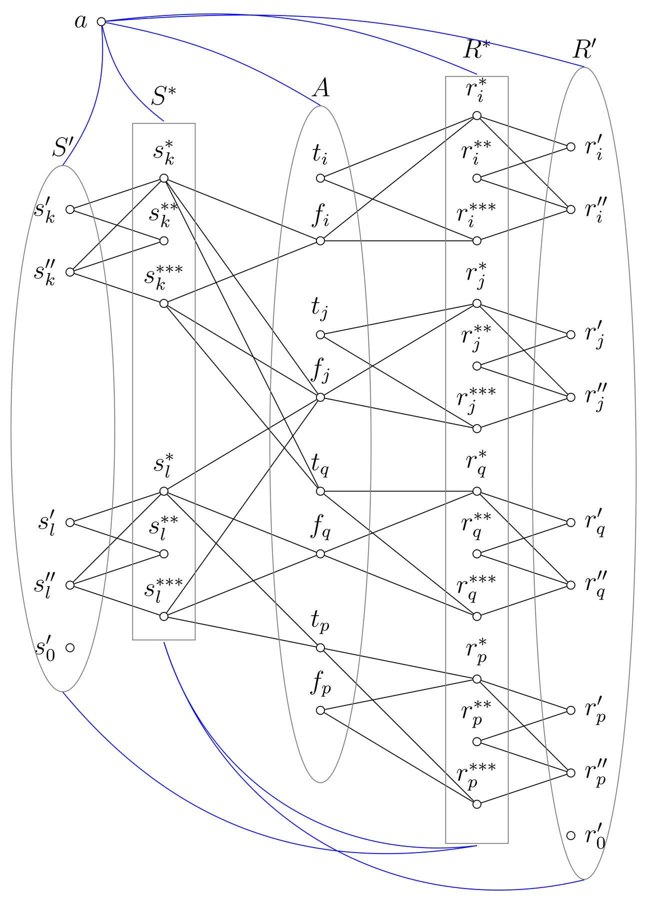
(2) improved near-tight running time upper and lower bounds for the problems on general graphs.

(3) fixed-parameter tractability for Non-Clash parameterized by the vertex integrity of G.

(4) lower bound excluding fixed-parameter tractability for Non-Clash parameterized by k plus the feedback vertex number and pathwidth of G.

Intractability and Running Time Bounds

Example: the graph G obtained by applying our reduction on the 3-SAT instance $\mathcal{X} = \{x_i, x_j, x_q, x_p\}$ and $\mathcal{C} = \{(x_i \vee x_j \vee \overline{x_q}), (x_j \vee x_q \vee \overline{x_p})\}$.



Ovals and rectangles depict stable sets and cliques, respectively. Blue edges represent all possible edges.

Positive non-clashing teaching map for $\mathcal{B}$.

for $k \in [m]$: $T(B_1(s_k^*)) = \{s_k', v\}$ and c_k is satisfied by $\tau(x_i)$ for some $i \in [n]$, where
- if $\tau(x_i) = \texttt{True}$, then $v = f_i$ $\quad (f_i \in B_1(s_k^*) \cap A$ since $x_i \in c_k)$
- if $\tau(x_i) = \texttt{False}$, then $v = t_i$ $\quad (t_i \in B_1(s_k^*) \cap A$ since $\overline{x_i} \in c_k)$

$T(B_1(s_k^{**})) = \{s_k', r_0'\}$ $\quad T(B_1(s_k^{***})) = \{s_k'', r_0'\}$

$T(B_1(s_k')) = \{s_k', s_k^{**}\}$ $\quad T(B_1(s_k'')) = \{s_k'', s_k^{***}\}$ $\quad T(B_1(s_0')) = \{s_0', a\}$

for $i \in [n]$: $T(B_1(r_i^*)) = \{r_i', v\}$, where
- if $\tau(x_i) = \texttt{True}$, then $v = t_i$
- if $\tau(f_i) = \texttt{False}$, then $v = f_i$

$T(B_1(r_i^{**})) = \{r_i', s_0'\}$ $\quad T(B_1(r_i^{***})) = \{r_i'', s_0'\}$

$T(B_1(r_i')) = \{r_i', r_i^{**}\}$ $\quad T(B_1(r_i'')) = \{r_i'', r_i^{***}\}$ $\quad T(B_1(r_0')) = \{r_0', a\}$

for $i \in [n]$: $T(B_1(t_i)) = \{t_i, a\}$ $\quad T(B_1(f_i)) = \{f_i, a\}$

$T(V(G)) = T(B_1(a)) = \{t_i, t_j\}$, for $i, j \in [n]$, $i \neq j$ such that there is no $k \in [m]$ where both x_i and x_j appear in c_k.

For each $u, z \in V(G)$, in a cell at the intersection of the corresponding row and column, we place a vertex $w \in V(G)$ such that $w \in T(B_1(u)) \cup T(B_1(z))$ and $w \notin B_1(u) \cup B_1(z)$.

	s_k^*	s_k^{**}	s_k^{***}	s_k'	s_k''	s_0'	t_i	f_i
r_i^*	v	r_0'	r_0'	r_i'	r_i'	r_i'	r_i'	r_i'
r_i^{**}	s_0'	r_0'	r_0'	s_0'	s_0'	r_i'	r_i'	r_i'
r_i^{***}	s_0'	r_0'	r_0'	s_0'	s_0'	r_i''	r_i''	r_i''
r_i'	s_k'	r_0'	r_0'	r_i'	r_i'	r_i'	r_i'	r_i'
r_i''	s_k'	r_0'	r_0'	r_i''	r_i''	r_i''	r_i''	r_i''
r_0'	s_k'	s_k'	s_k''	s_k'	s_k''	r_0'	r_0'	r_0'
t_j	s_k'	s_k'	s_k''	s_k'	s_k''	s_0'	t_i	f_i
f_j	s_k'	s_k'	s_k''	s_k'	s_k''	s_0'	t_i	f_i

	s_k^*	s_k^{**}	s_k^{***}	s_k'	s_k''
s_l^*	s_k'	v	s_k'	v	v
s_l^{**}	s_k'	s_k'	s_k'	r_0'	r_0'
s_l^{***}	s_k'	s_k'	s_k''	r_0'	r_0'
s_l'	s_k'	s_k'	s_k''	s_k'	s_k'
s_l''	s_k'	s_k'	s_k''	s_k'	s_k''
s_0'	s_0'	s_0'	s_0'	s_0'	s_0'

Here $i, j \in [n]$, $k \in [m]$, and $v \in \{t_i, f_i\} \setminus N(s_k^*)$: $\{t_i, f_i\} \setminus N(s_k^*) = t_i$ if $x_i = \texttt{True}$ satisfies c_k; and $\{t_i, f_i\} \setminus N(s_k^*) = f_i$ if $x_i = \texttt{False}$ satisfies c_k. For $k, l \in [m]$, filled cells correspond to the case $k = l$, and the others to $k \neq l$. Here, v is any vertex in $N(s_l^*) \cap A$. The table for vertices of the variable force-gadgets is defined similarly, interchanging all s and r symbols.

3-Satisfiability (3-SAT)

Input: A CNF formula over a set of clauses $\mathcal{C} = \{c_1, \ldots, c_m\}$ containing variables from $\mathcal{X} = \{x_1, \ldots, x_n\}$, where each clause has exactly 3 literals.

Question: Is there a variable assignment $\tau : \mathcal{X} \to \{\texttt{True}, \texttt{False}\}$ satisfying each clause in $\mathcal{C}$?

Theorem 1.

Strict Non-Clash is NP-hard even when restricted to the case of split graphs with $k = 2$.

Theorem 2.

Unless the *Exponential Time Hypothesis** fails, there is no algorithm solving Strict Non-Clash in time $2^{o(|V(G)| \cdot d \cdot k)}$, where d and k are the diameter of G and the target positive non-clashing teaching dimension of the instance, respectively.

* The *Exponential Time Hypothesis (ETH)*: There is a constant $c > 0$ such that there is no 2^{cn} algorithm for 3-SAT.
[Impagliazzo & Paturi, 2001]

Proposition 1.

Non-Clash can be solved in $2^{\mathcal{O}(|V(G)| \cdot d \cdot k \cdot \log |V(G)|)}$ time.

NAE-Integer-3-SAT

Input: A set of clauses $\mathcal{C}$ over variables $\mathcal{X}$, and an integer d. Any clause $c \in \mathcal{C}$ has the form $x \leq c_x$, $y \leq c_y$, and $z \leq c_z$, where $c_x, c_y, c_z \in \{1, \ldots, d\}$.

Question: Is there a variable assignment $\mathcal{X} \to \{1, \ldots, d\}$ such that for each clause, either one or two of its three inequalities are satisfied?

$\texttt{fvs}(G)$: the *feedback vertex number* of G is the cardinality of a smallest vertex subset $X \subset V(G)$ such that $G - X$ is acyclic.

$\texttt{pw}(G)$: the *pathwidth* of G has an involved definition based on the notion of *path decompositions*. However, it is well-known that deleting one vertex from each connected component of G will decrease the pathwidth by at most one, and that a graph consisting of a disjoint union of paths and *subdivided caterpillars* (*i.e.*, graphs consisting of a central path with pendent paths) has pathwidth 2.

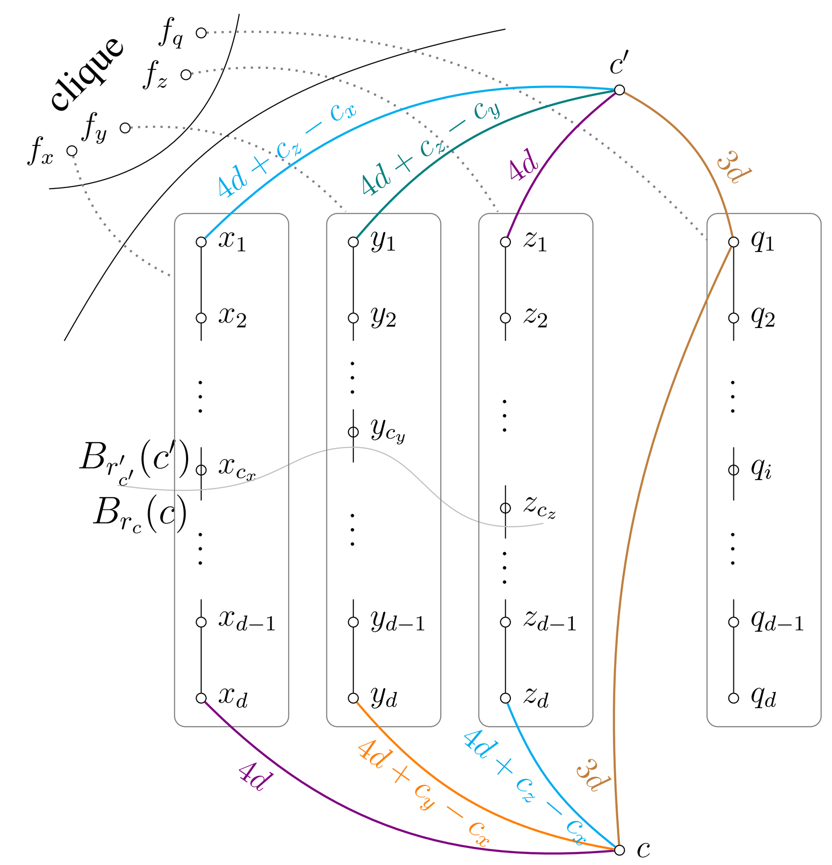
$\texttt{vi}(G)$: the *vertex integrity* of G is the smallest integer b such that there exists a vertex subset $X \subset V(G)$ with the property that, for every connected component H of $G - X$, $|V(H) \cup X| \leq b$.

Hardness for Classical Structural Parameterizations*

Theorem 3.

Non-Clash is W[1]-hard when parameterized by $\texttt{fvs}(G) + \texttt{pw}(G) + k$.

Example: the instance $(G, \mathcal{B}, k)$ of Non-Clash obtained by applying our reduction on the NAE-Integer-3-SAT formula.



Colorful edges denote paths of the depicted lengths. Two black curves separate the clique from the rest of the graph. Dotted edges show that a vertex of the clique is adjacent to all the vertices of the graph below the separating curves except those in the adjacent rectangle. The gray curve gives an intuition of which vertices of P_x, P_y, and P_z are contained in $B_{r_c}(c)$ and $B_{r'_{c'}}(c')$.

Fixed-Parameter Tractability* via Vertex Integrity

Theorem 4.

Non-Clash is FPT parameterized by the vertex integrity of the input graph G.

Consider an instance $(G, \mathcal{B}, k)$ of Non-Clash and let p be the vertex integrity of G.

Two subgraphs $H, H' \in \mathcal{H}$ are *twin-blocks* with respect to $\mathcal{B}$, denoted $H \sim_{\mathcal{B}} H'$, if there exists an isomorphism $\alpha_{H,H'}$ from H to H' with the following properties:
(1) for each $u \in V(H)$ and $v \in X$, $uv \in E(G)$ if and only if $\alpha_{H,H'}(u)v \in E(G)$, and
(2) for each $u \in V(H)$ and $r \in \mathbb{N}$, $B_r(u) \in \mathcal{B}$ if and only if $B_r(\alpha_{H,H'}(u)) \in \mathcal{B}$.

Proof Sketch.

Compute the witness $X \subset V(G)$ for the vertex integrity and the corresponding set $\mathcal{H}$ of connected components.

Classify the elements of $\mathcal{H}$ w.r.t. the equivalence classes defined by $\sim_{\mathcal{B}}$.

Observation 1.
There are at most $2^{\mathcal{O}(p^3)}$ equivalence classes, and the equivalence between two components can be tested in $p^{\mathcal{O}(p)}$ time,

Lemma 1.
The equivalence classes can be computed in $|V(G)| \cdot 2^{\mathcal{O}(p^3)} \cdot p^{\mathcal{O}(p)}$ time with brute force.

Compute the equivalent reduced graph G' of G by removing some components of $\mathcal{H}$ whose equivalence class is larger than some $f(p)$. This is the crux of the algorithm, which is made possible by carefully analyzing hypothetical solutions.

Lemma 2.
The set $\mathcal{B}'$ induced by $\mathcal{B}$ on G' can be computed in $|V(G')| \cdot \mathcal{O}(p^2) \cdot |V(G)| \cdot |V(G')|$ time.

The size of the instance $(G', \mathcal{B}', k)$ is a function of p. Thus, we can compute a positive non-clashing teaching map of dimension at most k for $\mathcal{B}'$ in time depending only on p.

* In *parameterized complexity*, the running-times of algorithms are studied with respect to a parameter $p \in \mathbb{N}$ and input size n. A parameterized problem is *fixed-parameter tractable* (FPT) if it can be solved by an algorithm running in time $f(p) \cdot n^{\mathcal{O}(1)}$, where f is a computable function.
[Downey & Fellows, 1999; Cygan et al., 2015]

Concluding Remarks

One open question highlighted by our work concerns the tiny remaining gap between the algorithmic lower and upper bounds obtained in Theorem 2 and Proposition 1. In particular, is there a way to improve the running time of the latter algorithm to $2^{\mathcal{O}(|V(G)| \cdot d \cdot k)}$ and make the bounds tight?
General directions for future work are to perform a similar complexity analysis in the setting where negative examples are allowed, and to consider approximability.