PREPROCESSING COMPLEXITY FOR SOME GRAPH PROBLEMS PARAMETERIZED BY STRUCTURAL PARAMETERS

Manuel Lafond,

Weidong Luo

Université de Sherbrooke, Canada

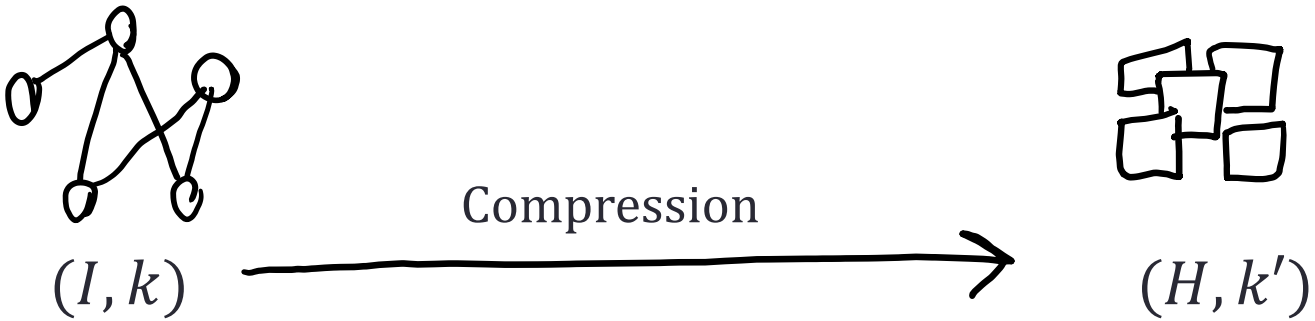
PREPROCESSING ON GRAPHS USING STRUCTURAL PARAMETERS

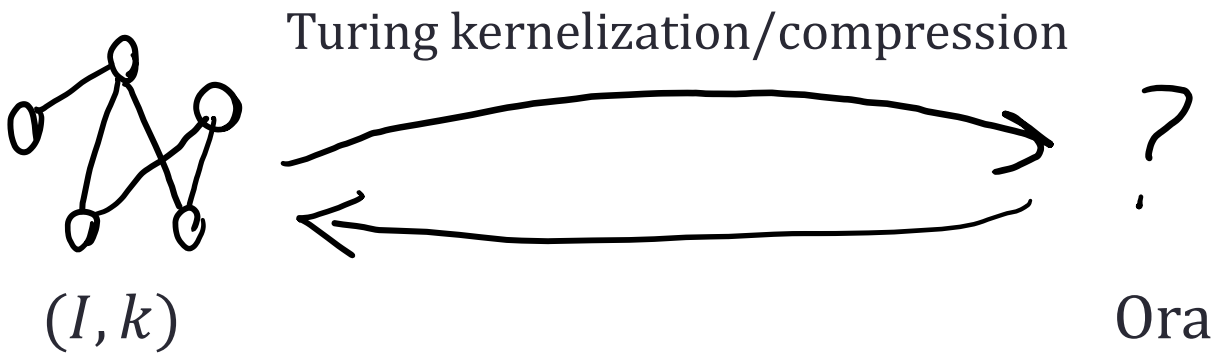
Manuel Lafond,

Weidong Luo

Université de Sherbrooke, Canada


Kernelization
(I, k)
(I', k')


Compression
(I, k)
(H, k')


Turing kernelization/compression
?
(I, k)
Oracle

Kernelization for problem Q

- polytime algorithm on instance (I, k) of Q
- produces instance (I', k') of Q
- $(I, k) \in Q \Leftrightarrow (I', k') \in Q$
- $|I'|, k' \in O(f(k))$ for some function f that depends only on k

Good when $f(k) \in O(k^c)$. If this holds, we have a polynomial kernel.

Kernelization for problem Q

- polytime algorithm on instance (I, k) of Q
- produces instance (I', k') of Q
- $(I, k) \in Q \Leftrightarrow (I', k') \in Q$
- $|I'|, k' \in O(f(k))$ for some function f that depends only on k

Good when $f(k) \in O(k^c)$. If this holds, we have a polynomial kernel.

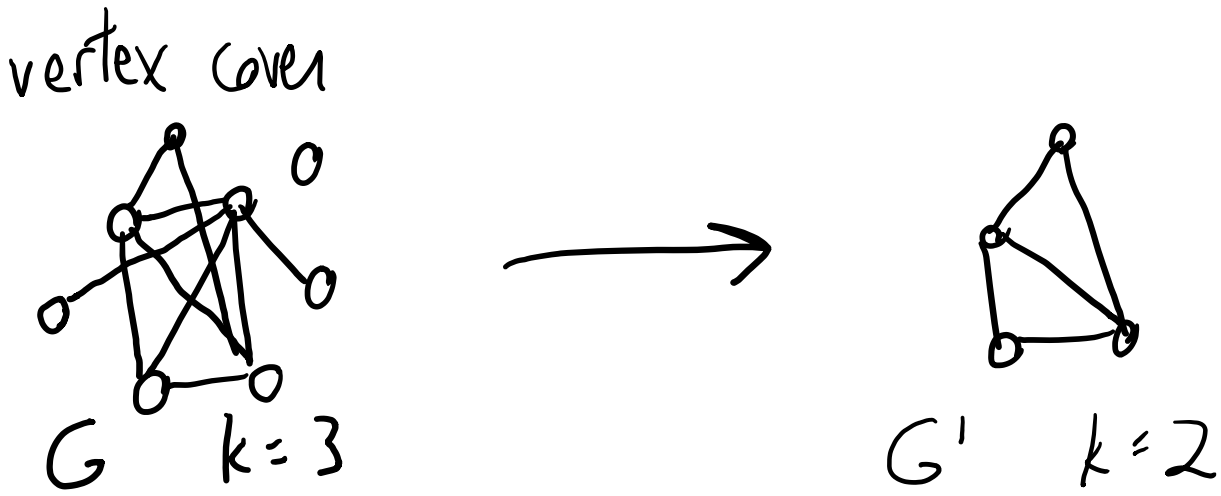
Example:

- Vertex cover problem:
- Given graph G and parameter k
- Does there exist a vertex cover of size at most k?

Reduction rules:

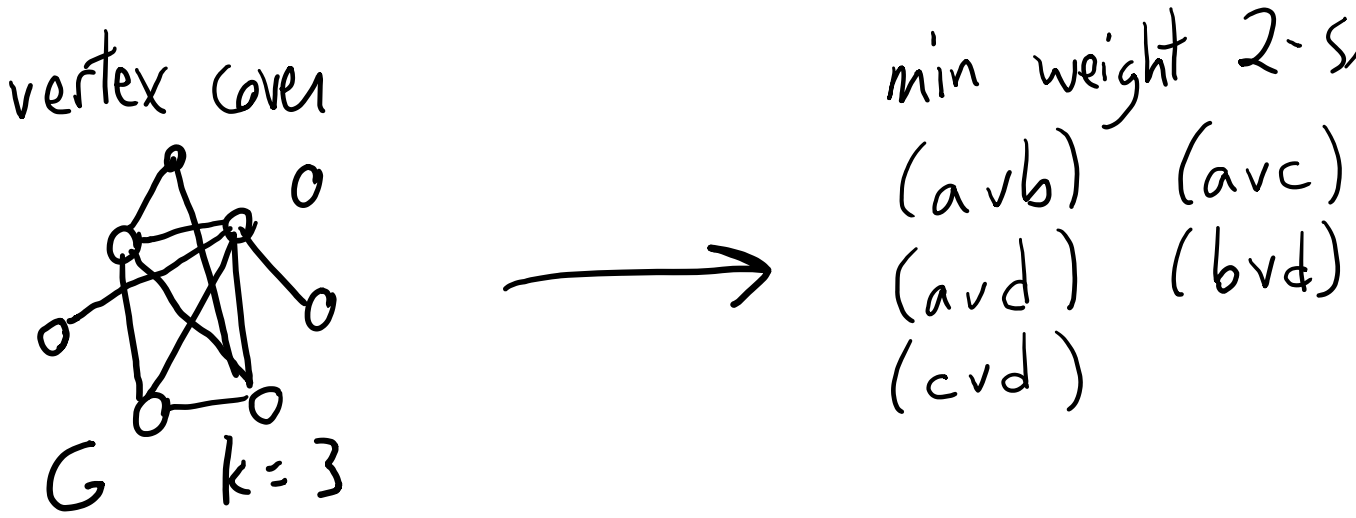
- Delete isolated vertices
- Delete vertices of degree $> k$, reduce k by 1

Preserve an equivalent instance of size $O(k^2)$.



Compression for problem Q

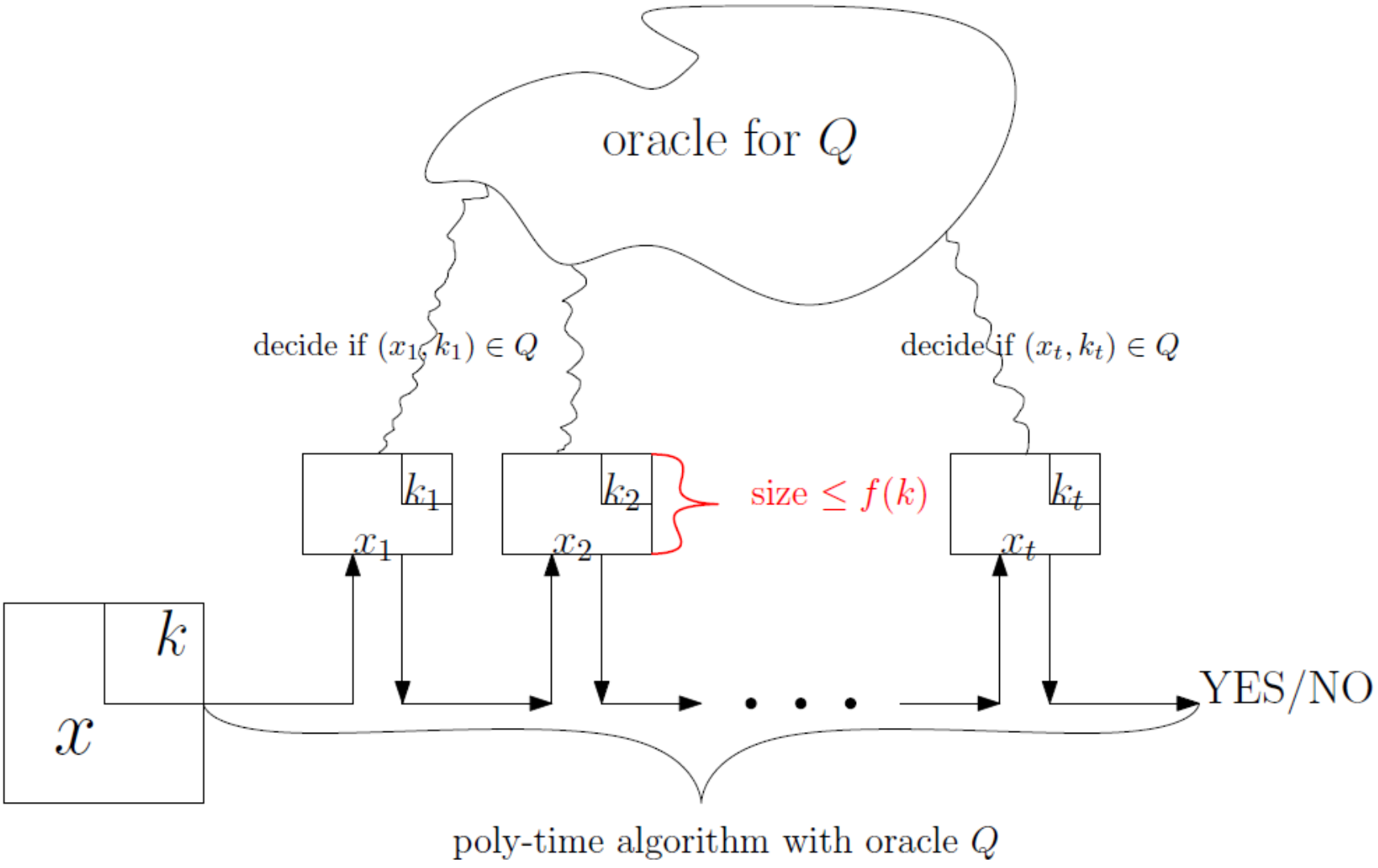
- polytime algorithm on instance (I, k) of Q
- produces instance (I', k') of **some other problem** R
- $(I, k) \in Q \Leftrightarrow (I', k') \in R$
- $|I'|, k' \in O(f(k))$ for some function f that depends only on k



Turing kernelization for problem Q

- polytime algorithm on instance (I, k) of Q
- can query an Oracle O for instances (I', k') of Q where $|I'|, k \in O(f(k))$

If query sizes are $O(k^c)$, we have a polynomial Turing kernel.


oracle for Q
decide if $(x_1, k_1) \in Q$
decide if $(x_t, k_t) \in Q$
k_1
x_1
k_2
x_2
size $\leq f(k)$
k_t
x_t
k
x
YES/NO
poly-time algorithm with oracle Q

Turing compression for problem Q

- polytime algorithm on instance (I, k) of Q
- can query an Oracle O for instances (I', k') of R where $|I'|, k \in O(f(k))$

PK = polynomial kernel

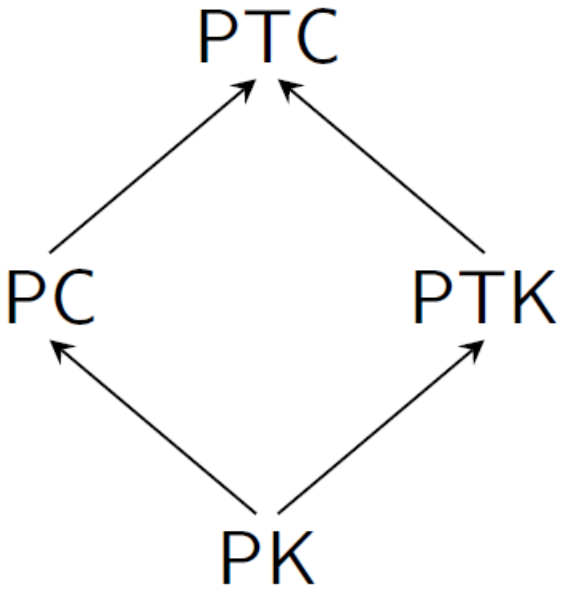
PC = polynomial compression

PTK = polynomial Turing kernel

PTC = polynomial Turing compression

WK-hard = probably no PK

MK-hard = probably no PK



Natural parameter: k = value to optimize.

- Vertex cover admits a polynomial kernel of size $2k$, where k is the size of the cover.
- Clique (probably) does not admit a kernel of any size wrt k, where k is the size of the clique.

Structural parameter: kernelize w.r.t. some graph invariant.

- Max-Clique admits a kernel of size $2^{vc} + vc$, where vc is the size of a vertex cover of G.
- Max-Clique admits a Turing kernel of query size $vc + 1$. (explained later)

In this paper

What is the preprocessing complexity of algorithmic problems, parameterized by:

- vertex cover (vc)
- twin-cover (tc)
- neighborhood diversity (nd)
- modular-width (mw)

For problem Q, we write e.g. $Q(vc)$ to indicate that we parameterize by vc.

Problems \ Parameters	vc	tc	nd	mw
Partition Into Triangle	PK	PK	PC	open no PC [24]
Induced Matching	WK[1]-h	WK[1]-h	PC	PTC [29] no PC
Steiner Tree	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Chromatic Number	MK[2]-h	MK[2]-h	PC	PTC [29] no PC
Connected Dominating Set	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Dominating Set	MK[2]-h [21]	MK[2]-h [21]	PC [19]	PTC [29] no PC
Hamiltonian Cycle	PK [4]	PK [4]	PC	PTC [29] no PC
Clique	PTK [5] no PC [5]	PTK no PC [5]	PC [19]	PTC [29] no PC
Independent Set	PK [8]	PK	PC [19]	PTC [29] no PC
Vertex Cover	PK [8]	PK	PC [19]	PTC [29] no PC
Feedback Vertex Set	PK [22]	PK [18]	PC [19]	PTC [29] no PC
Odd Cycle Transversal	PK [23]	PK	PC [19]	PTC [29] no PC
Connected Vertex Cover	WK[1]-h [21]	WK[1]-h [21]	PC [19]	PTC [29] no PC

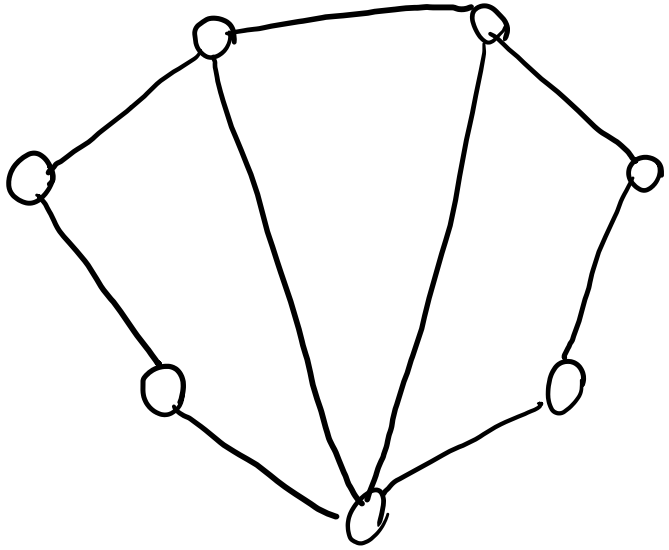
Problems \ Parameters	vc	tc	nd	mw
Partition Into Triangle	PK	PK	PC	open no PC [24]
Induced Matching	WK[1]-h	WK[1]-h	PC	PTC [29] no PC
Steiner Tree	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Chromatic Number	MK[2]-h	MK[2]-h	PC	PTC [29] no PC
Connected Dominating Set	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Dominating Set	MK[2]-h [21]	MK[2]-h [21]	PC [19]	PTC [29] no PC
Hamiltonian Cycle	PK [4]	PK [4]	PC	PTC [29] no PC
Clique	PTK [5] no PC [5]	PTK no PC [5]	PC [19]	PTC [29] no PC
Independent Set	PK [8]	PK	PC [19]	PTC [29] no PC
Vertex Cover	PK [8]	PK	PC [19]	PTC [29] no PC
Feedback Vertex Set	PK [22]	PK [18]	PC [19]	PTC [29] no PC
E Transversal	PK [23]	PK	PC [19]	PTC [29] no PC
D er	WK[1]-h [21]	WK[1]-h [21]	PC [19]	PTC [29] no PC

= Weidong

= Manuel

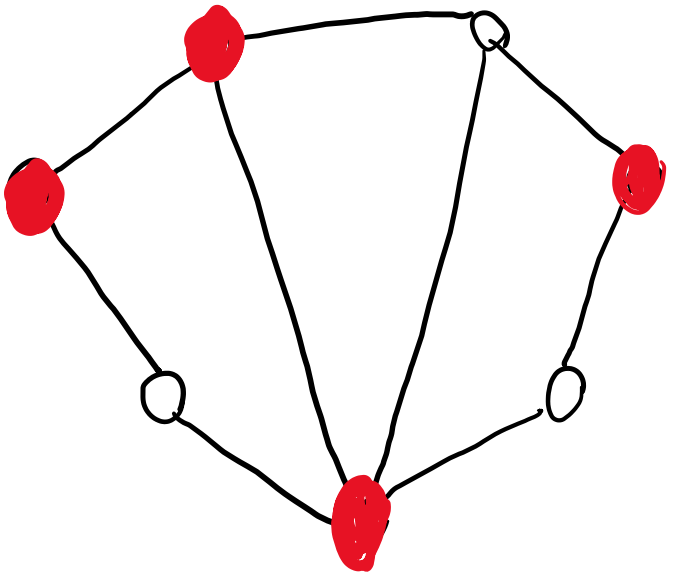
Vertex cover - $vc(G)$ (or just vc)

vc is the minimum size of a subset of vertices that touches every edge.



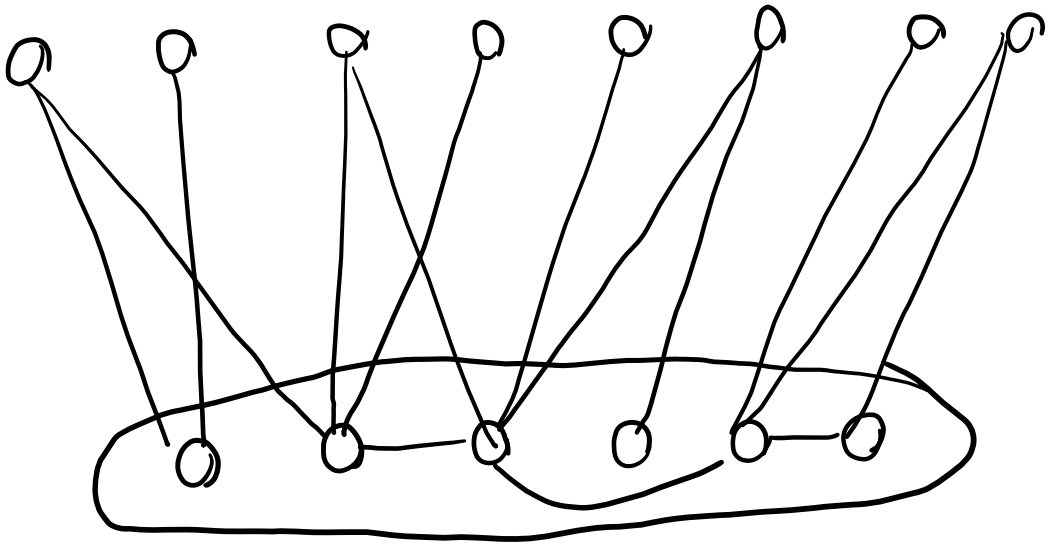
Vertex cover - $vc(G)$ (or just vc)

vc is the minimum size of a subset of vertices that touches every edge.



Vertex cover - $vc(G)$ (or just vc)

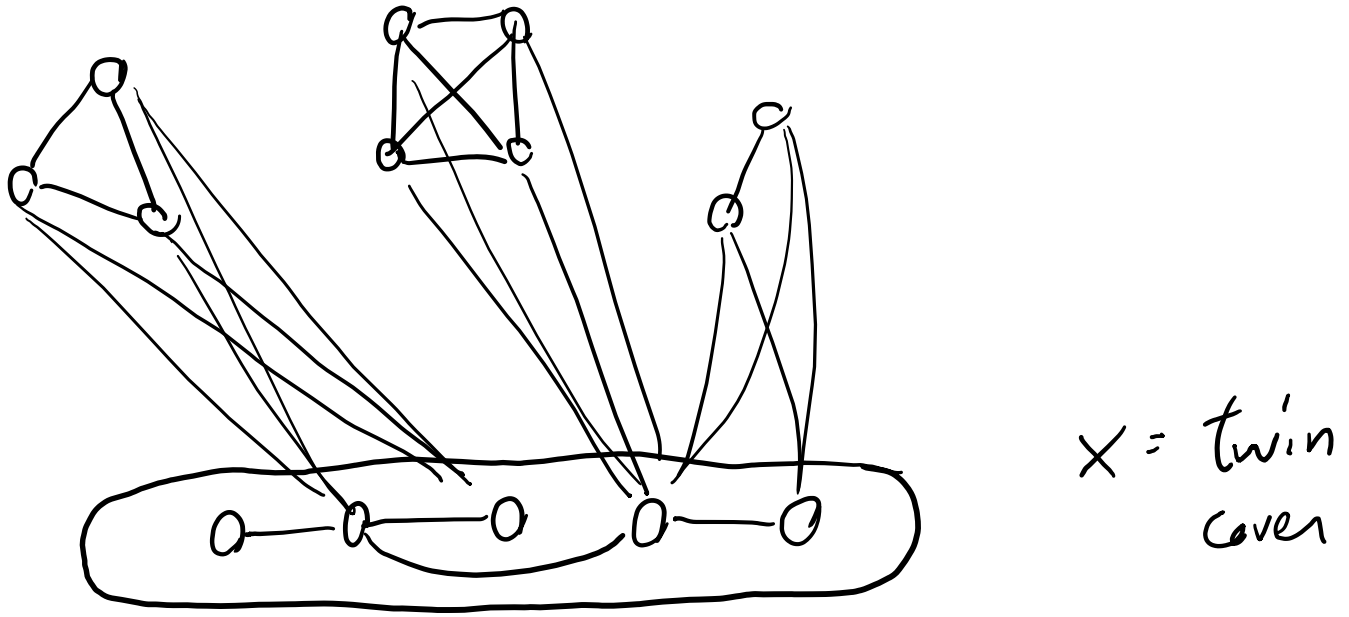
vc is the minimum size of a subset of vertices that touches every edge.



Twin cover - $tc(G)$

Introduced in [Ganian 2011].

tc is the minimum size of a subset $X \subseteq V(G)$ such that $G - X$ is a cluster graph (contains disconnected cliques), and vertices in the same clique of $G - X$ are twins.



Twin cover - $tc(G)$

Introduced in [Ganian 2011].

tc is the minimum size of a subset $X \subseteq V(G)$ such that $G - X$ is a cluster graph (contains disconnected cliques), and vertices in the same clique of $G - X$ are twins.

Similar to vc, except that every outside vertex is blown up into a clique.

$tc \leq vc$

Problems \ Parameters	vc	tc	nd	mw
Partition Into Triangle	PK	PK	PC	open no PC [24]
Induced Matching	WK[1]-h	WK[1]-h	PC	PTC [29] no PC
Steiner Tree	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Chromatic Number	MK[2]-h	MK[2]-h	PC	PTC [29] no PC
Connected Dominating Set	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Dominating Set	MK[2]-h [21]	MK[2]-h [21]	PC [19]	PTC [29] no PC
Hamiltonian Cycle	PK [4]	PK [4]	PC	PTC [29] no PC
Clique	PTK [5] no PC [5]	PTK no PC [5]	PC [19]	PTC [29] no PC
Independent Set	PK [8]	PK	PC [19]	PTC [29] no PC
Vertex Cover	PK [8]	PK	PC [19]	PTC [29] no PC
Feedback Vertex Set	PK [22]	PK [18]	PC [19]	PTC [29] no PC
Odd Cycle Transversal	PK [23]	PK	PC [19]	PTC [29] no PC
Connected Vertex Cover	WK[1]-h [21]	WK[1]-h [21]	PC [19]	PTC [29] no PC

Max-Clique(tc)

- Max-Clique does not admit a polynomial compression in parameter vc unless coNP $\subseteq$ NP$_{/\text{poly}}$ [Bodlaender, Jansen, Kratsch, 2014]
  - No poly kernel in vc nor tc either.

Proposition

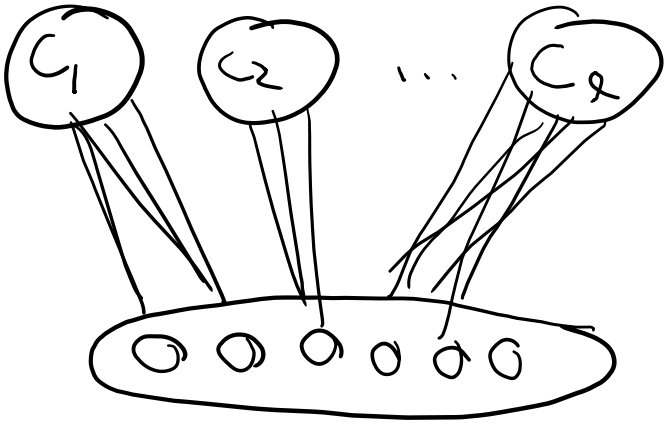
Max-Clique admits a polynomial Turing kernel in parameter tc.

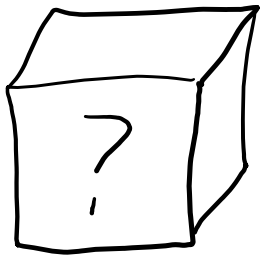
Proposition

Max-Clique admits a polynomial Turing kernel in parameter tc.

Let X such that $G - X$ is a cluster graph, each clique containing twins. For each clique C in $G - X$:

- Take any $c \in C$, ask Oracle for Max-Clique in $G[X + c]$ (query size $tc + 1$).
- If Oracle returns k, then using C we can have a clique of size $k + |C| - 1$ (some details skipped)



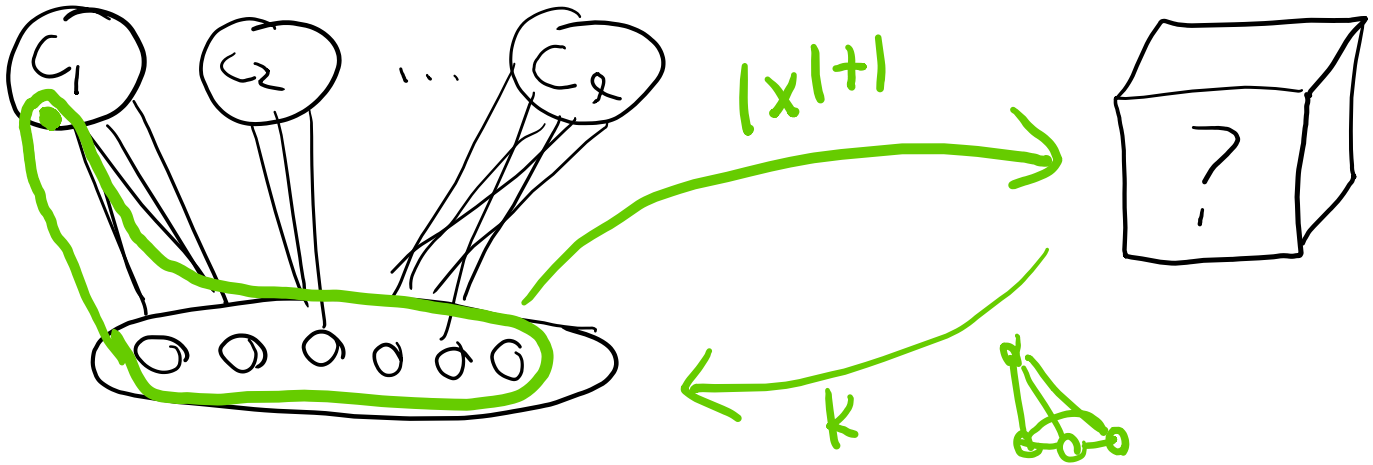


Proposition

Max-Clique admits a polynomial Turing kernel in parameter tc.

Let X such that $G - X$ is a cluster graph, each clique containing twins. For each clique C in $G - X$:

- Take any $c \in C$, ask Oracle for Max-Clique in $G[X + c]$ (query size $tc + 1$).
- If Oracle returns k, then using C we can have a clique of size $k + |C| - 1$ (some details skipped)

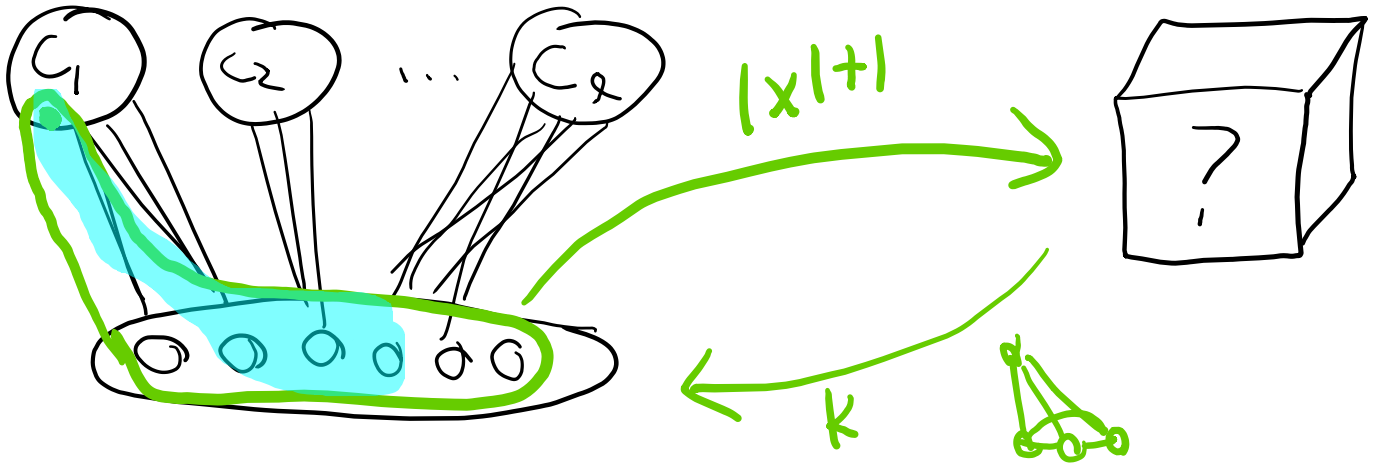


Proposition

Max-Clique admits a polynomial Turing kernel in parameter tc.

Let X such that $G - X$ is a cluster graph, each clique containing twins. For each clique C in $G - X$:

- Take any $c \in C$, ask Oracle for Max-Clique in $G[X + c]$ (query size $tc + 1$).
- If Oracle returns k, then using C we can have a clique of size $k + |C| - 1$ (some details skipped)

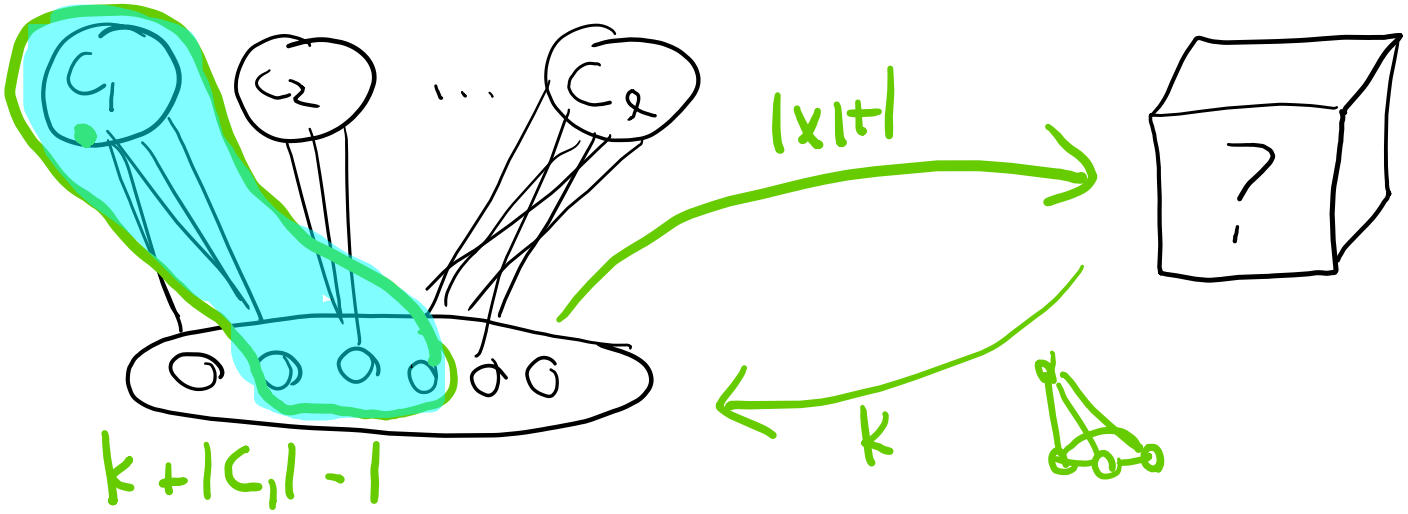


Proposition

Max-Clique admits a polynomial Turing kernel in parameter tc.

Let X such that $G - X$ is a cluster graph, each clique containing twins. For each clique C in $G - X$:

- Take any $c \in C$, ask Oracle for Max-Clique in $G[X + c]$ (query size $tc + 1$).
- If Oracle returns k, then using C we can have a clique of size $k + |C| - 1$ (some details skipped)

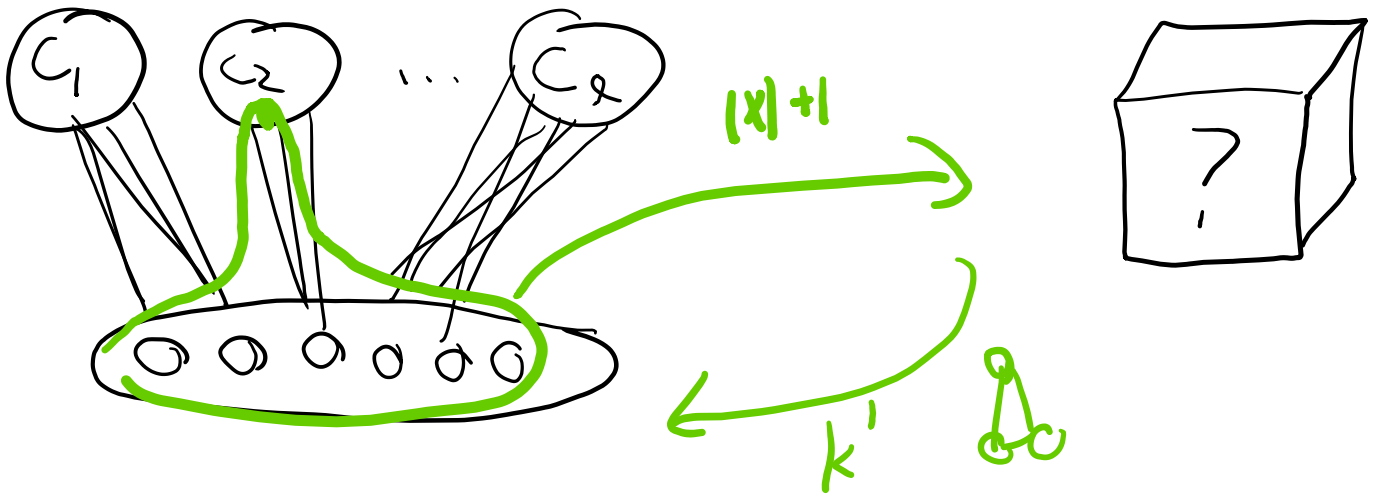


Proposition

Max-Clique admits a polynomial Turing kernel in parameter tc.

Let X such that $G - X$ is a cluster graph, each clique containing twins. For each clique C in $G - X$:

- Take any $c \in C$, ask Oracle for Max-Clique in $G[X + c]$ (query size $tc + 1$).
- If Oracle returns k, then using C we can have a clique of size $k + |C| - 1$ (some details skipped)



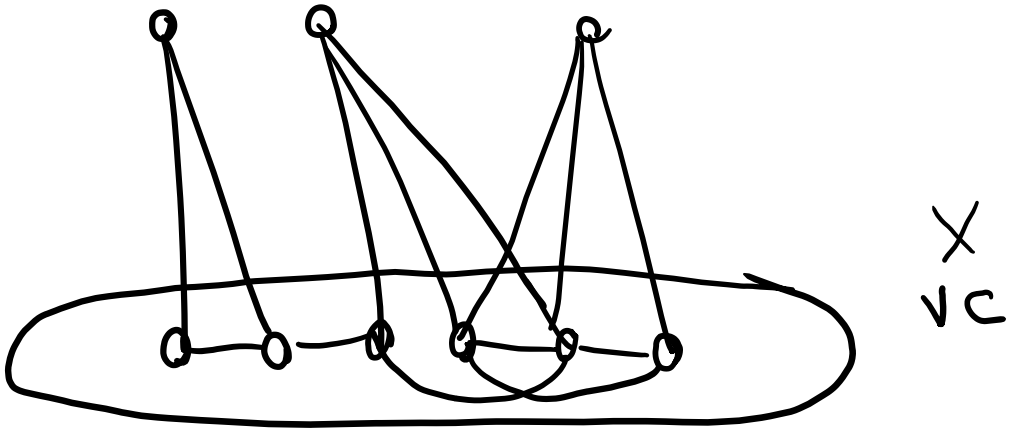
Problems \ Parameters	vc	tc	nd	mw
Partition Into Triangle	PK	PK	PC	open no PC [24]
Induced Matching	WK[1]-h	WK[1]-h	PC	PTC [29] no PC
Steiner Tree	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Chromatic Number	MK[2]-h	MK[2]-h	PC	PTC [29] no PC
Connected Dominating Set	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Dominating Set	MK[2]-h [21]	MK[2]-h [21]	PC [19]	PTC [29] no PC
Hamiltonian Cycle	PK [4]	PK [4]	PC	PTC [29] no PC
Clique	PTK [5] no PC [5]	PTK no PC [5]	PC [19]	PTC [29] no PC
Independent Set	PK [8]	PK	PC [19]	PTC [29] no PC
Vertex Cover	PK [8]	PK	PC [19]	PTC [29] no PC
Feedback Vertex Set	PK [22]	PK [18]	PC [19]	PTC [29] no PC
Odd Cycle Transversal	PK [23]	PK	PC [19]	PTC [29] no PC
Connected Vertex Cover	WK[1]-h [21]	WK[1]-h [21]	PC [19]	PTC [29] no PC

Triangle-Partition

Input: graph G

Output: is there a partition of $V(G)$ into $n/3$ cliques of size 3?

Triangle-Partition(vc): trivial $3/2 \cdot vc$ kernel.

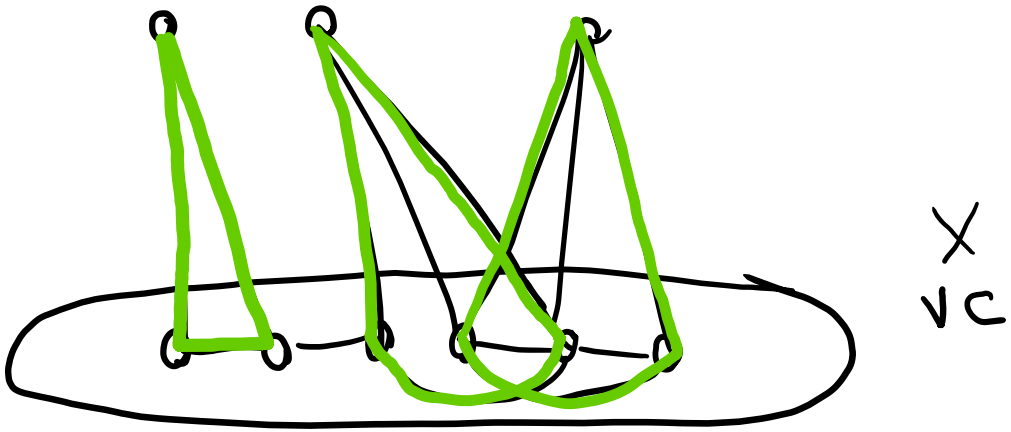


Triangle-Partition

Input: graph G

Output: is there a partition of $V(G)$ into $n/3$ cliques of size 3?

Triangle-Partition(vc): trivial $3/2 \cdot vc$ kernel.

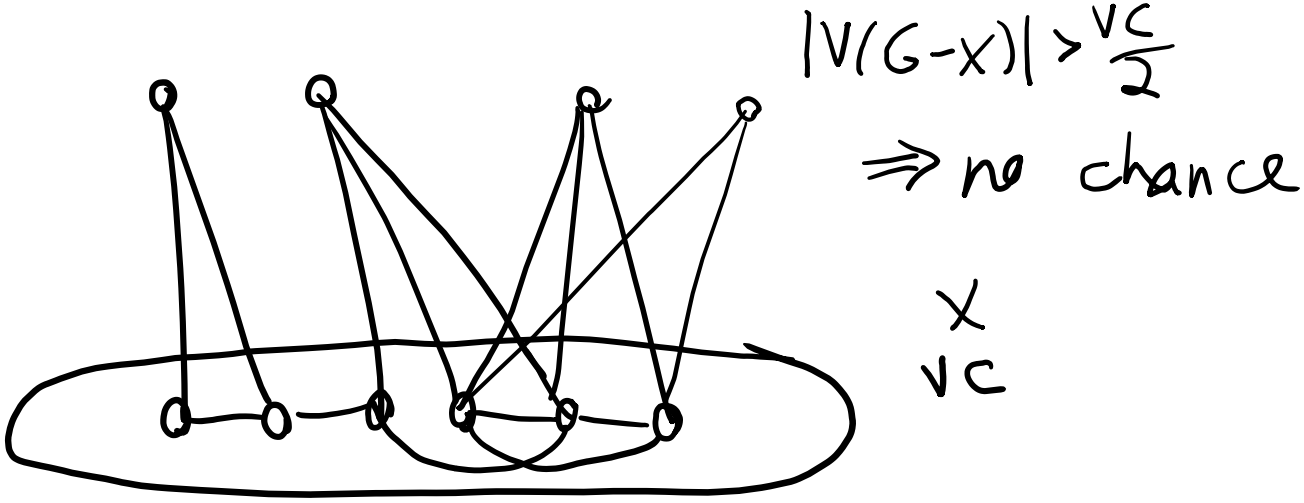


Triangle-Partition

Input: graph G

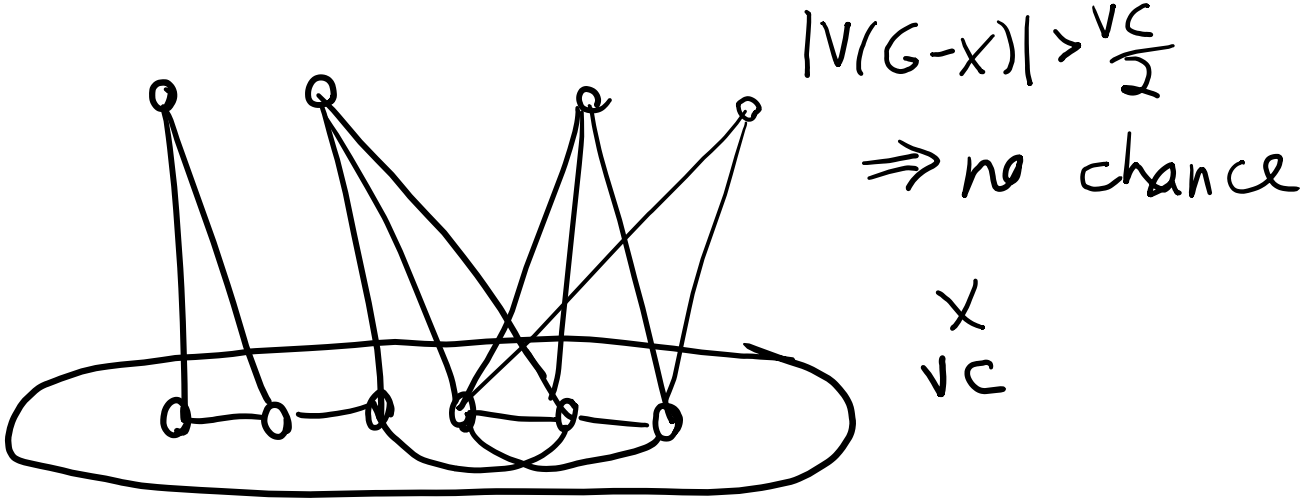
Output: is there a partition of $V(G)$ into $n/3$ cliques of size 3?

Triangle-Partition(vc): trivial $3/2 \cdot vc$ kernel.



Triangle-Partition(vc): trivial $3/2 \cdot vc$ kernel.

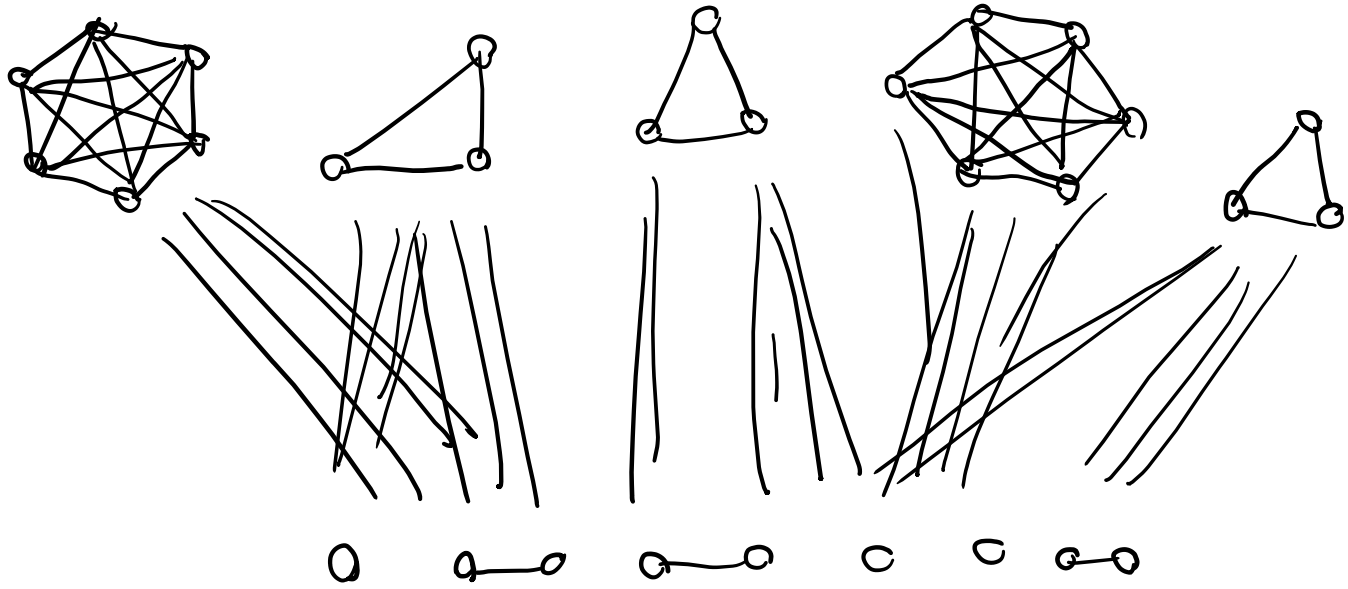
Let X be a vertex cover of G of size vc. Then $G - X$ is an independent set. Each vertex of $G - X$ must be in a triangle with two vertices of X. Thus if $|V(G - X)| > vc/2$, we say NO, otherwise $|V(G)| = |X| + |V(G - X)| \leq 3/2vc$.

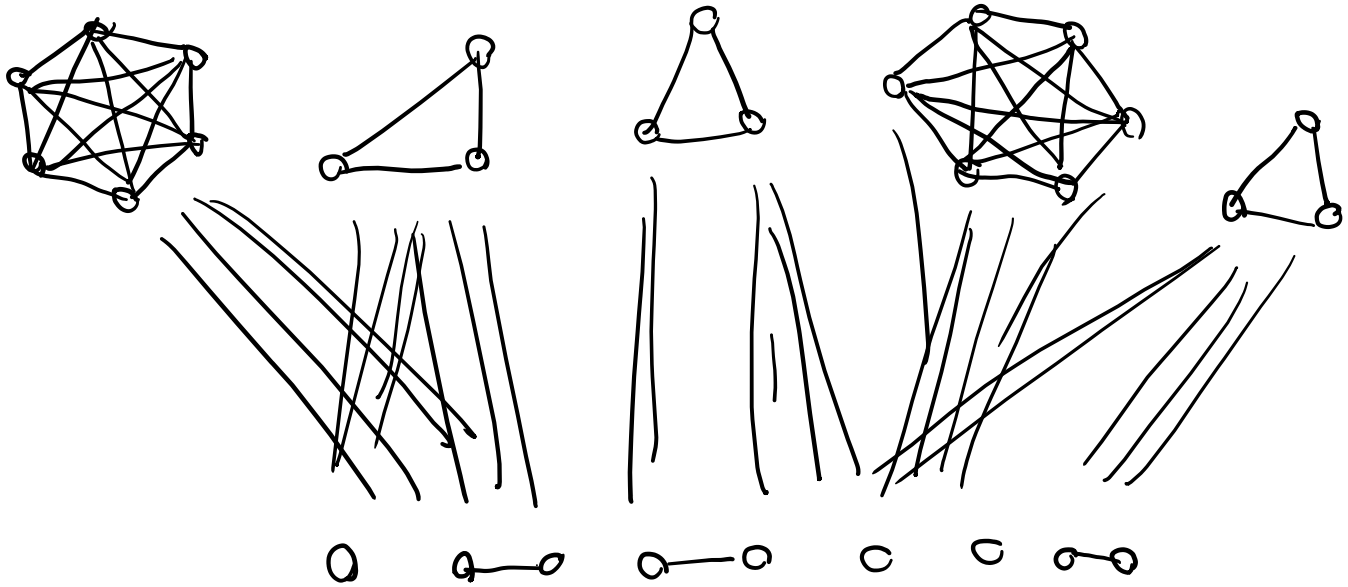


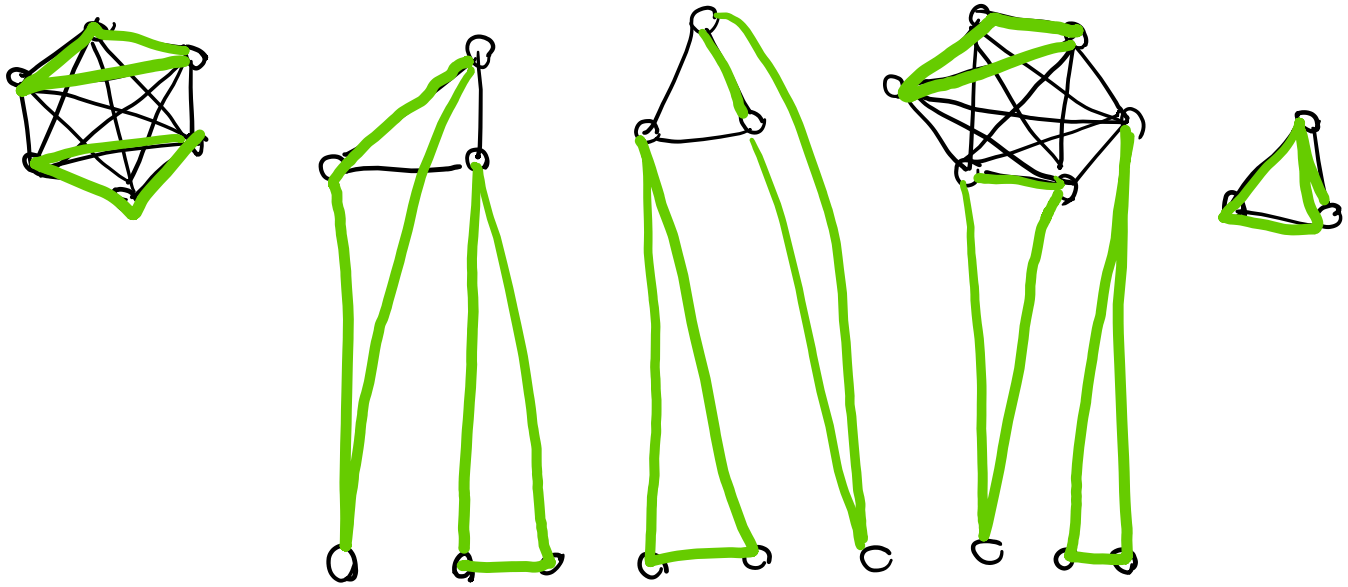
Triangle-Partition(tc): much, much less trivial.

Let $X \subseteq V(G)$ such that $G - X$ is a cluster graph. There can be many cliques in $G - X$ of size 0 (mod 3), that are "self-sufficient". But we don't know which ones are, and which ones use X.

Polynomial compression to a new problem called Conflict Free Assignment.



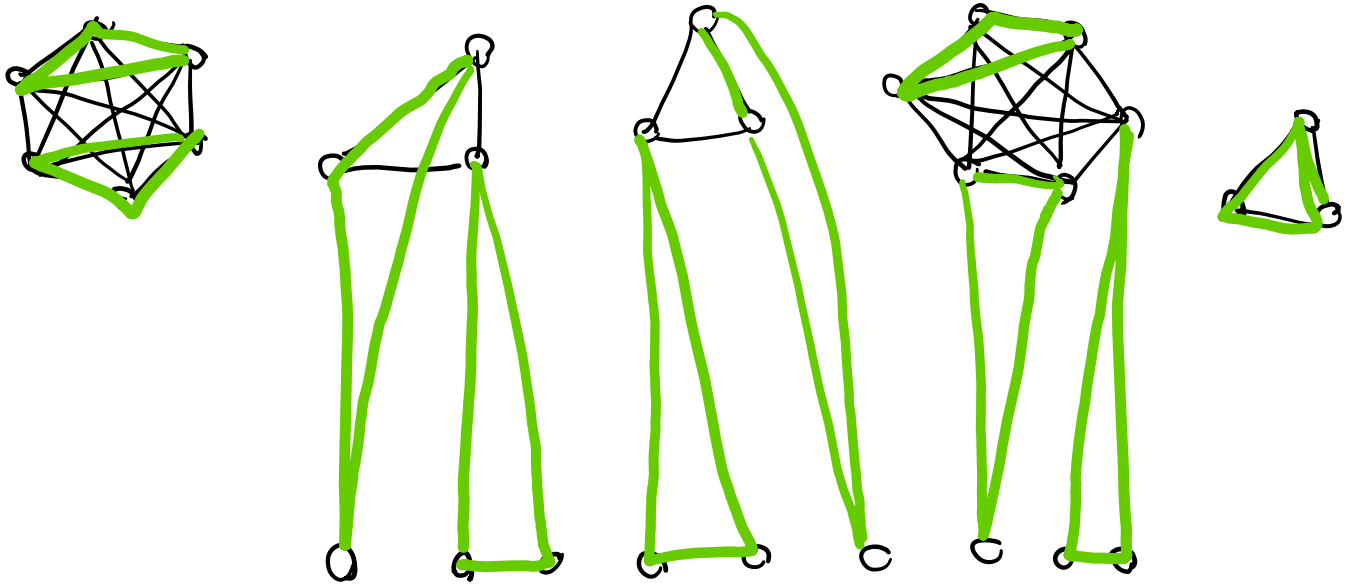




Triangle-Partition(tc): much, much less trivial.

Let $X \subseteq V(G)$ such that $G - X$ is a cluster graph. There can be many cliques in $G - X$ of size 0 (mod 3), that are "self-sufficient". But we don't know which ones are, and which ones use X.

Polynomial compression to a new problem called Conflict Free Assignment.



Conflict-Free Assignment

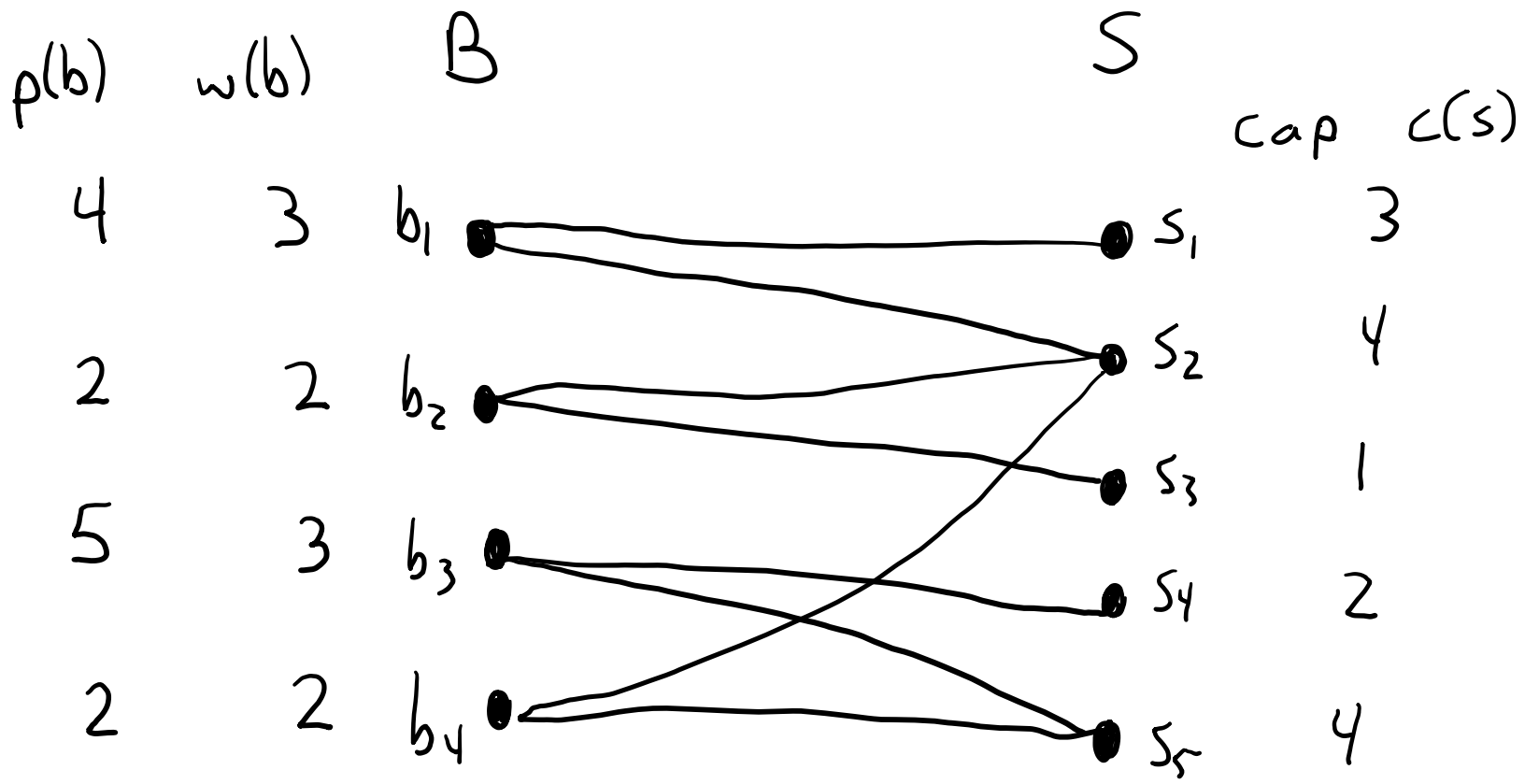
Input:

- bipartite graph $G = (B \cup S, E)$, $B =$ buyers $S =$ sellers.
- each buyer $b \in B$ has a weight $w(b)$ and a profit $p(b)$.
- each seller $s \in S$ has a capacity $c(s)$.
- $P \subseteq \binom{B}{2}$ forbidden pairs of buyers.

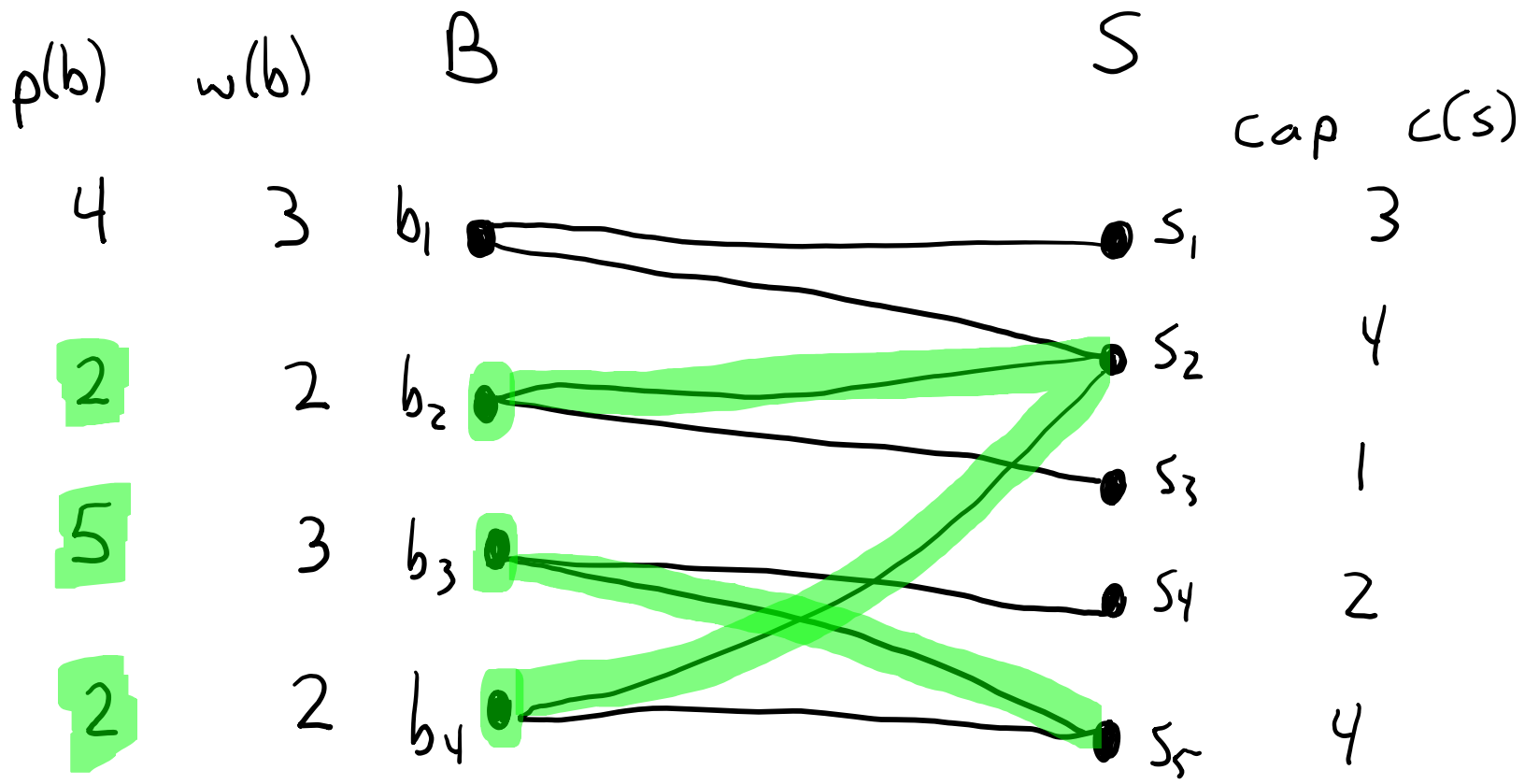
Output:

Partial assignment of buyers to sellers such that

- buyer b assigned to s only if $\{b, s\} \in E$.
- sum of weights $w(b)$ assigned to seller s is at most $c(s)$.
- if $\{b, b'\} \in P$, then one of b or b' is not assigned anywhere.
- sum of profit of assigned buyers is maximum.



$P = \{ \{b_1, b_3\} \}$



$P = \{ \{b_1, b_3\} \}$

Profit $= 2+5+2 = 9$

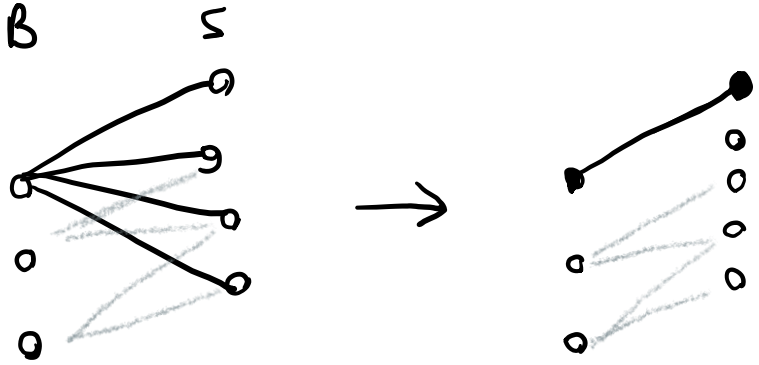
Theorem

The Conflict-Free Assignment problem admits a polynomial kernel in parameter $|B|$, the number of buyers.

Idea: if some buyer b has $\geq |B| + 1$ neighbors in S, there will be a seller available for b no matter what.

So, just remove all neighbors of b, add a new seller s_b whose sole neighbor is b.

As a result, each buyer has at most $|B|$ neighbors, hence we have a kernel of size $O(|B|^2)$.



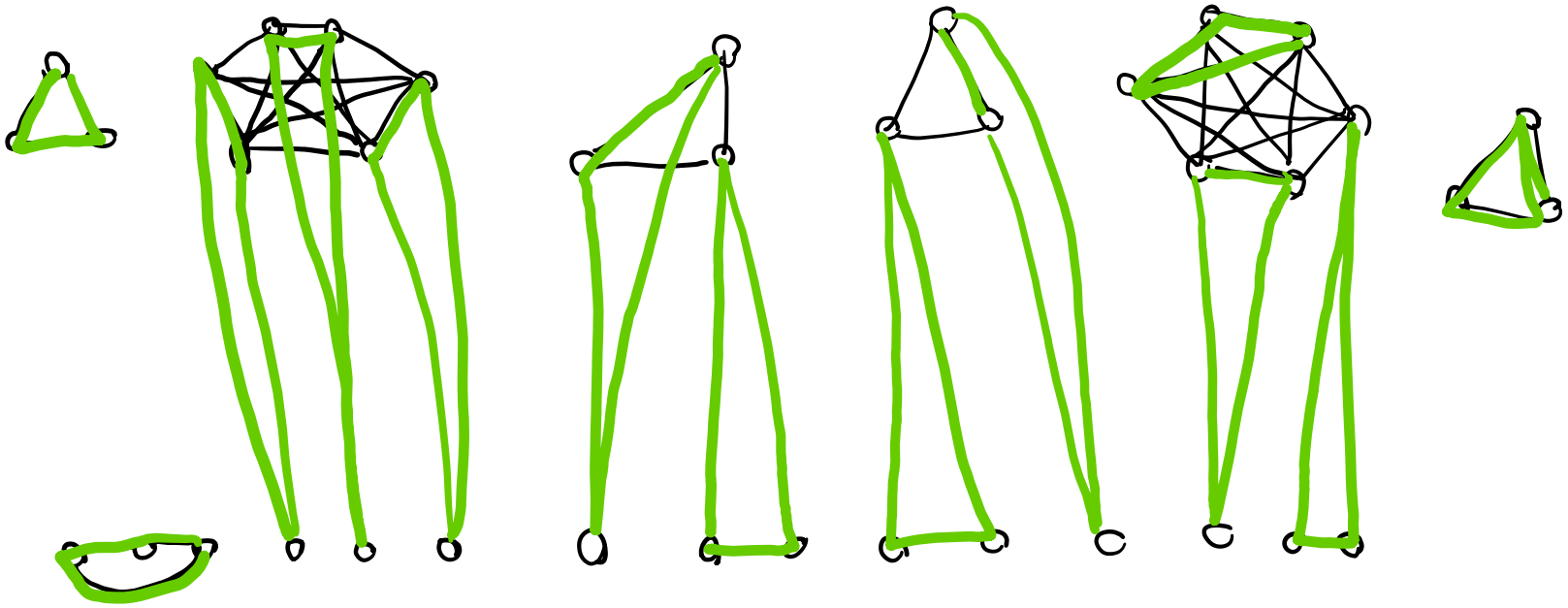
* Delete isolated vertices

Compression TrianglePartition(tc) to Conflict-Free Assignment

Assume that each clique of $G - X$ has size 0 (mod 3).

Then each clique "uses" 0 (mod 3) vertices of X to make triangles.

Challenge: assign triples of X to their rightful clique if any.

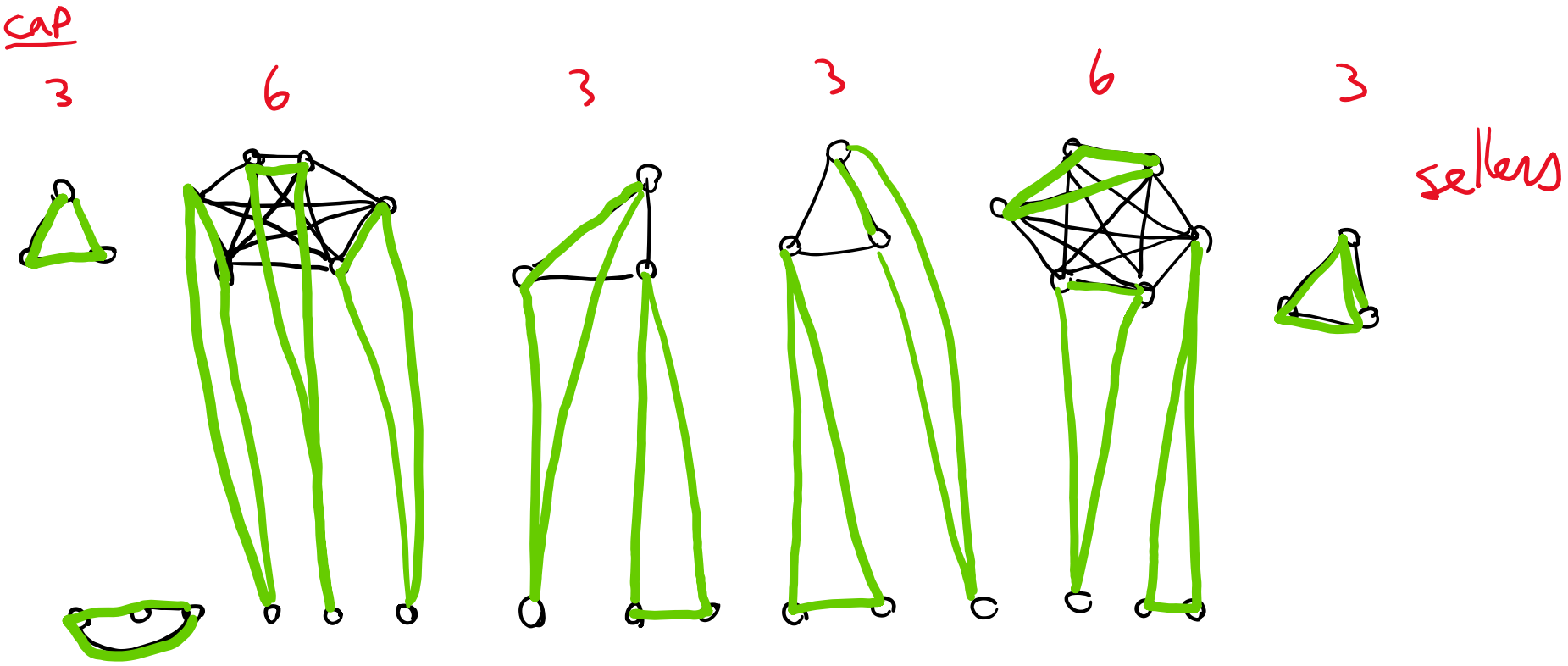


Compression TrianglePartition(tc) to Conflict-Free Assignment

Assume that each clique of $G - X$ has size 0 (mod 3).

Then each clique "uses" 0 (mod 3) vertices of X to make triangles.

Challenge: assign triples of X to their rightful clique if any.

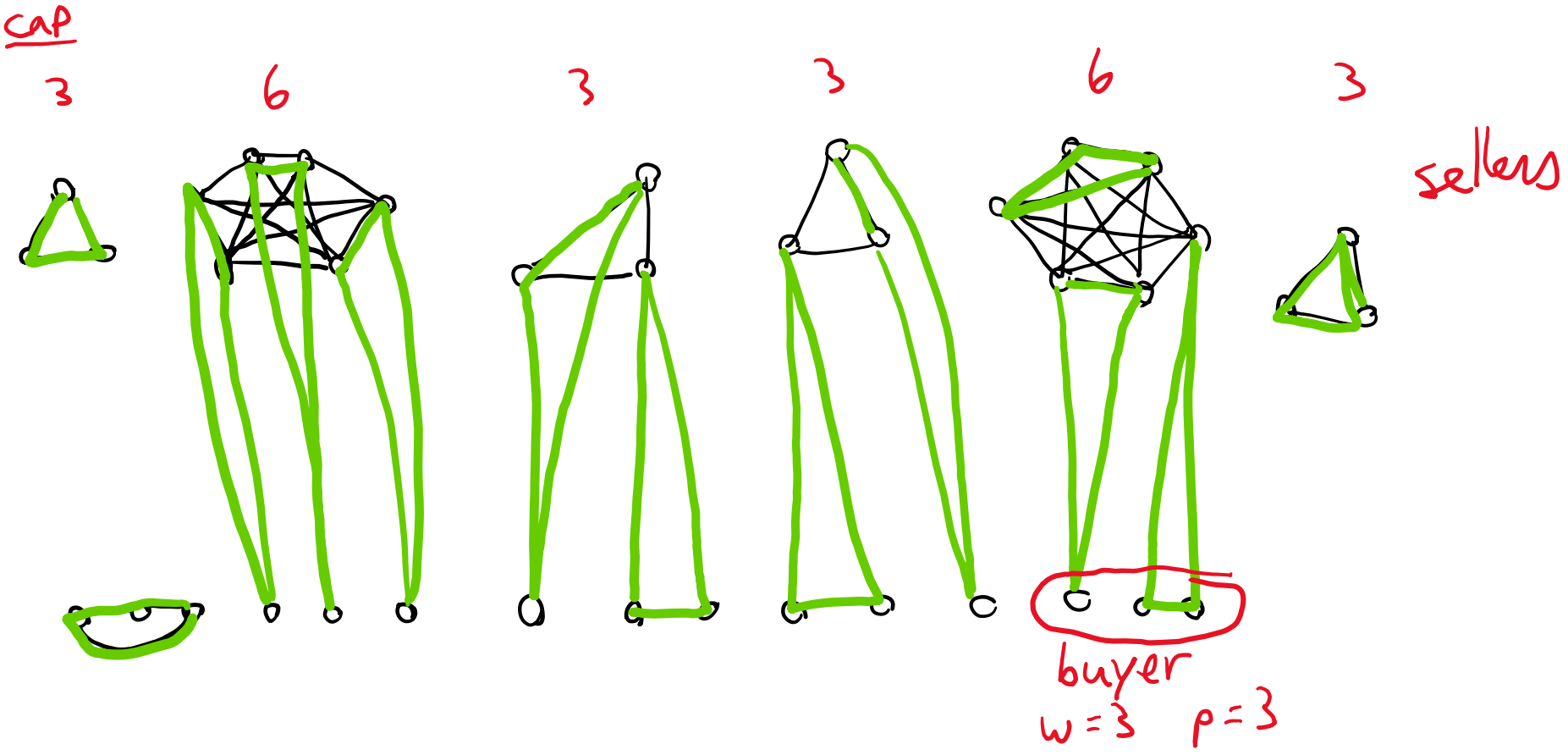


Compression TrianglePartition(tc) to Conflict-Free Assignment

Assume that each clique of $G - X$ has size 0 (mod 3).

Then each clique "uses" 0 (mod 3) vertices of X to make triangles.

Challenge: assign triples of X to their rightful clique if any.

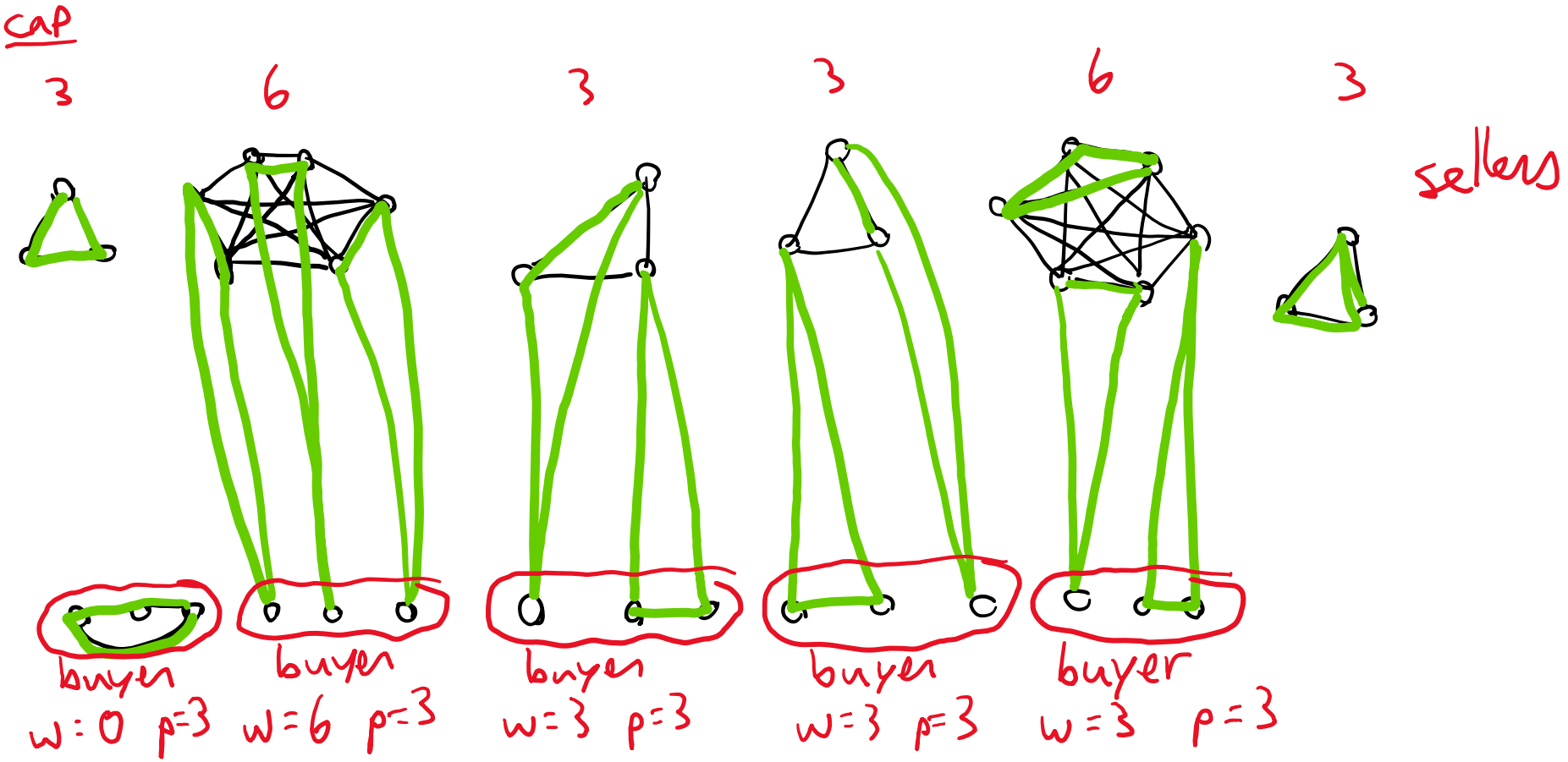


Compression TrianglePartition(tc) to Conflict-Free Assignment

Assume that each clique of $G - X$ has size 0 (mod 3).

Then each clique "uses" 0 (mod 3) vertices of X to make triangles.

Challenge: assign triples of X to their rightful clique if any.



Compression TrianglePartition(tc) to Conflict-Free Assignment

Assume that each clique of $G - X$ has size 0 (mod 3).

Then each clique "uses" 0 (mod 3) vertices of X to make triangles.

Challenge: assign triples of X to their rightful clique if any.

Buyers: all triples $t \in \binom{X}{3}$ of X. Each has profit 3.

Forbidden pairs: pair of triples $\{t_1, t_2\}$ with $t_1 \cap t_2 \neq \emptyset$.

Weight of triple = number of vertices it needs in a cluster to form triangles. (0, 3, or 6)

Sellers = clusters. Capacity $c(C) = |C|$

Exists triangle partition iff exists assignment of profit |X|

Compression TrianglePartition(tc) to Conflict-Free Assignment

Assume that each clique of $G - X$ has size 0 (mod 3).

Then each clique "uses" 0 (mod 3) vertices of X to make triangles.

Challenge: assign triples of X to their rightful clique if any.

Buyers: all triples $t \in \binom{X}{3}$ of X. Each has profit 3. *

Forbidden pairs: pair of triples $\{t_1, t_2\}$ with $t_1 \cap t_2 \neq \emptyset$.

Weight of triple = number of vertices it needs in a cluster to form triangles. (0, 3, or 6)

Sellers = clusters. Capacity $c(C) = |C|$

Exists triangle partition iff exists assignment of profit |X|

* in paper, $t \in \binom{X}{3} \cup \binom{X}{6}$

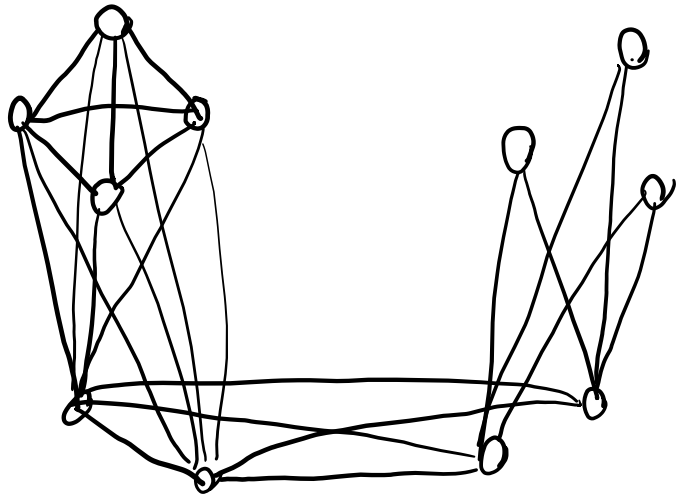
Problems \ Parameters	vc	tc	nd	mw
Partition Into Triangle	PK	PK	PC	open no PC [24]
Induced Matching	WK[1]-h	WK[1]-h	PC	PTC [29] no PC
Steiner Tree	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Chromatic Number	MK[2]-h	MK[2]-h	PC	PTC [29] no PC
Connected Dominating Set	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Dominating Set	MK[2]-h [21]	MK[2]-h [21]	PC [19]	PTC [29] no PC
Hamiltonian Cycle	PK [4]	PK [4]	PC	PTC [29] no PC
Clique	PTK [5] no PC [5]	PTK no PC [5]	PC [19]	PTC [29] no PC
Independent Set	PK [8]	PK	PC [19]	PTC [29] no PC
Vertex Cover	PK [8]	PK	PC [19]	PTC [29] no PC
Feedback Vertex Set	PK [22]	PK [18]	PC [19]	PTC [29] no PC
Odd Cycle Transversal	PK [23]	PK	PC [19]	PTC [29] no PC
Connected Vertex Cover	WK[1]-h [21]	WK[1]-h [21]	PC [19]	PTC [29] no PC

Neighborhood diversity - $nd(G)$

Introduced in [Lampis, 2010]

A *module* of G is a subset $M \subseteq V$ such that every vertex of M has the same neighborhood outside of M.

nd is the minimum size of a partition of $V(G)$ into modules that each induce a clique or an independent set.

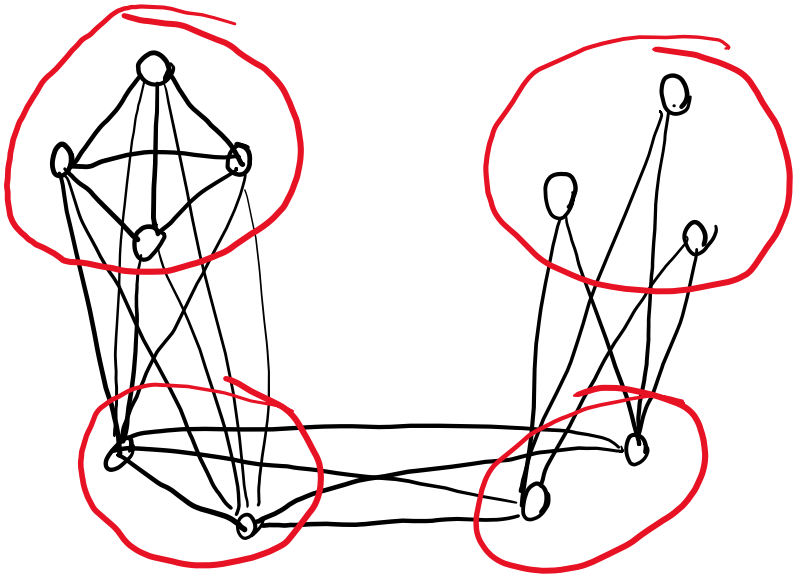


Neighborhood diversity - $nd(G)$

Introduced in [Lampis, 2010]

A *module* of G is a subset $M \subseteq V$ such that every vertex of M has the same neighborhood outside of M.

nd is the minimum size of a partition of $V(G)$ into modules that each induce a clique or an independent set.

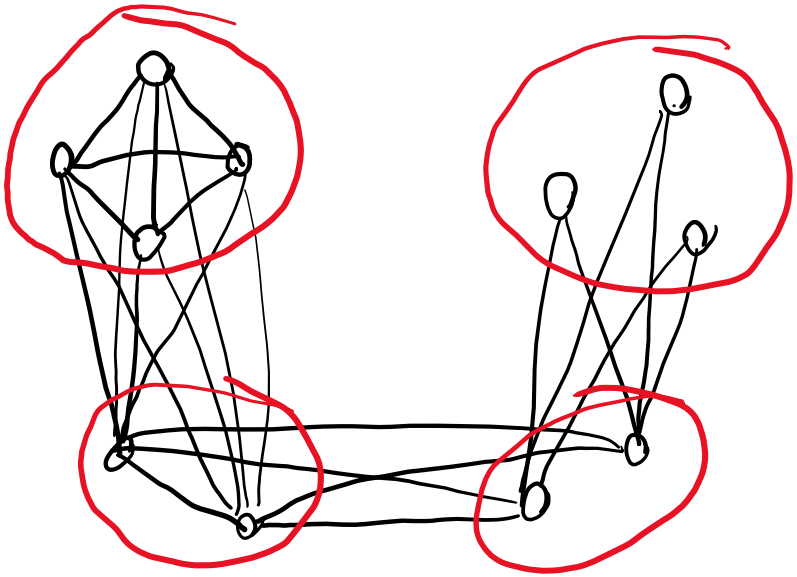


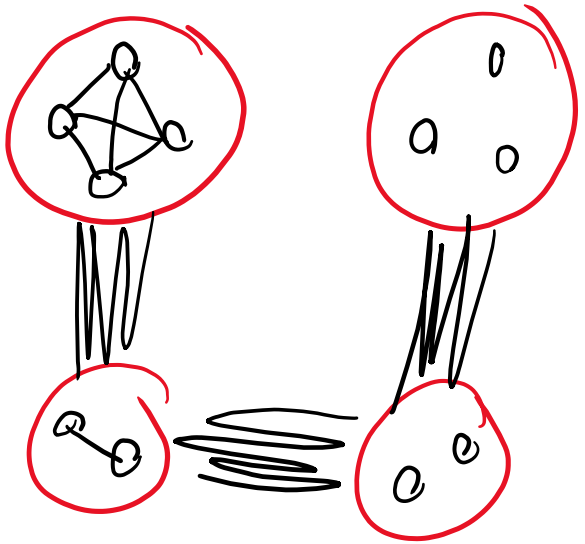
Neighborhood diversity - $nd(G)$

Introduced in [Lampis, 2010]

A *module* of G is a subset $M \subseteq V$ such that every vertex of M has the same neighborhood outside of M.

nd is the minimum size of a partition of $V(G)$ into modules that each induce a clique or an independent set.



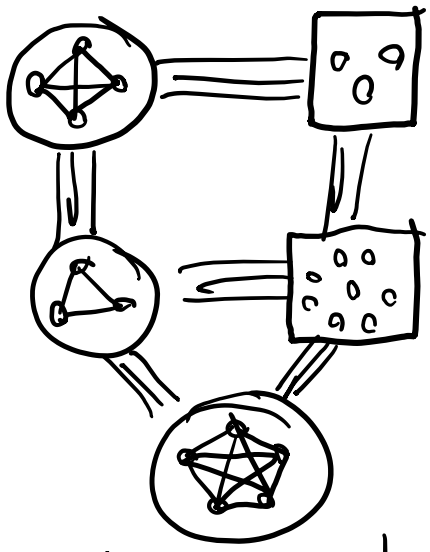


Kernels in nd

Proposition

If Q is a graph problem that can be solved in time $2^{O(nd^c)}$, then Q admits a polynomial compression.

Idea: if $nd^c < \log n$, solve in polytime $2^{O(nd^c)} = 2^{O(\log n)}$.
Assume that $nd^c \geq \log n$.



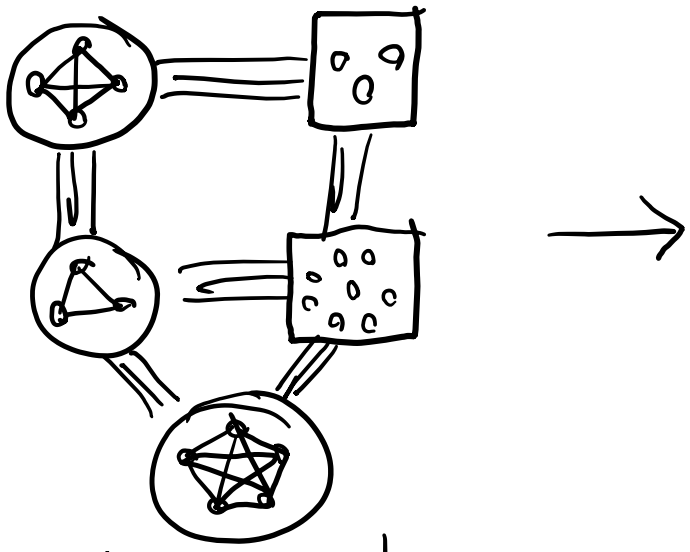
nd modules, n vertices

Kernels in nd

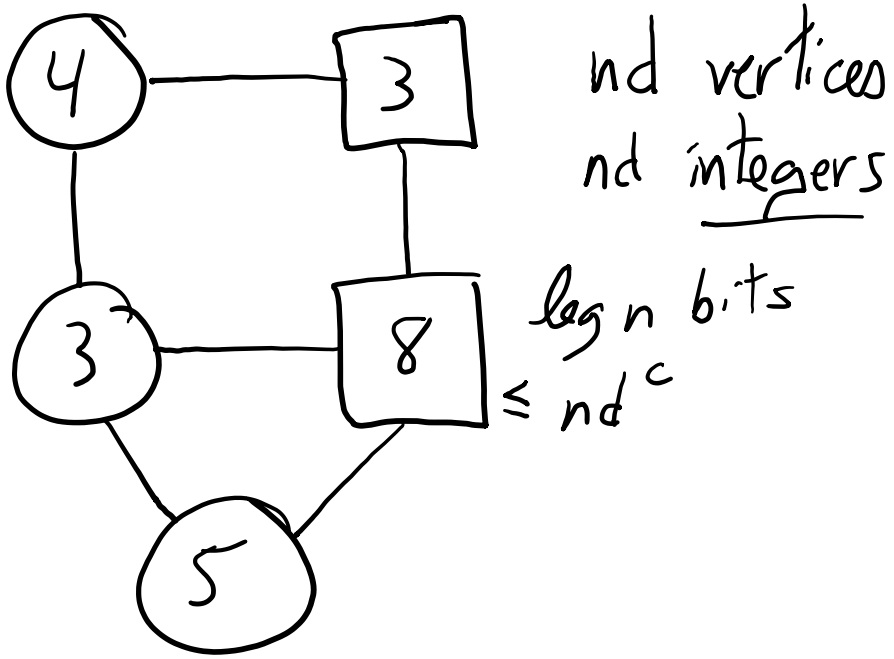
Proposition

If Q is a graph problem that can be solved in time $2^{O(nd^c)}$, then Q admits a polynomial compression.

Idea: if $nd^c < \log n$, solve in polytime $2^{O(nd^c)} = 2^{O(\log n)}$.
Assume that $nd^c \geq \log n$.



nd modules, n vertices



Kernels in nd

Proposition

If Q is a graph problem that can be solved in time $2^{O(nd^c)}$, then Q admits a polynomial compression.

Idea: if $nd^c < \log n$, solve in polytime $2^{O(nd^c)} = 2^{O(\log n)}$.
Assume that $nd^c \geq \log n$.

For each clique module M, encode it as $(1, |M|)$.

For each ind. set. module M, encode it as $(0, |M|)$.

Represent graph with nd vertices, each encoded.

Total size is at most $nd \cdot \log n \leq nd^{c+1}$.

We have compressed into the problem

"same problem as Q but the size of the modules are given as integers".

Kernels in nd

Proposition

If Q is a graph problem that can be solved in time $2^{O(nd^c)}$, then Q admits a polynomial compression.

Since every graph property expressible in MSO_1 can be solved in $2^{O(\mathsf{nd})}|G|^{O(1)}$ [27], the result of Lemma 8 covers problems outside MSO_1. The following problem can be solved in $2^{O(\mathsf{nd})}n^{O(1)}$ time: Hamiltonian Cycle [1, 26, 27], Connected Vertex Cover [2], Connected Dominating Set [2], Steiner Tree [2], Dominating Set [7], Odd Cycle Transversal [29], Vertex Cover [14], Independent Set [14], Clique [14], Feedback Vertex Set [14], Induced Matching [14], and Chromatic Number [16], where the facts that $\mathsf{nd}(G) \leq \mathsf{cw}(G) + 1$ [27] and $\mathsf{mw}(G) \leq \mathsf{nd}(G)$ are needed when some results of the references are used. In addition, Partition Into Tirangle [24] can be solved in $O^*(2^{\mathsf{nd}^{O(1)}})$ time.

Problems \ Parameters	vc	tc	nd	mw
Partition Into Triangle	PK	PK	PC	open no PC [24]
Induced Matching	WK[1]-h	WK[1]-h	PC	PTC [29] no PC
Steiner Tree	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Chromatic Number	MK[2]-h	MK[2]-h	PC	PTC [29] no PC
Connected Dominating Set	MK[2]-h	MK[2]-h	PK [29]	PK [29]
Dominating Set	MK[2]-h [21]	MK[2]-h [21]	PC [19]	PTC [29] no PC
Hamiltonian Cycle	PK [4]	PK [4]	PC	PTC [29] no PC
Clique	PTK [5] no PC [5]	PTK no PC [5]	PC [19]	PTC [29] no PC
Independent Set	PK [8]	PK	PC [19]	PTC [29] no PC
Vertex Cover	PK [8]	PK	PC [19]	PTC [29] no PC
Feedback Vertex Set	PK [22]	PK [18]	PC [19]	PTC [29] no PC
Odd Cycle Transversal	PK [23]	PK	PC [19]	PTC [29] no PC
Connected Vertex Cover	WK[1]-h [21]	WK[1]-h [21]	PC [19]	PTC [29] no PC

Modular-width – $mw(G)$

A module of G is a $M \subseteq V$ such that every vertex of M has the same neighborhood outside of M.

∅, V, and every {v} for v ∈ V are called the trivial modules.

If all modules of G are trivial modules then G is a prime graph.

mw is the number of vertices in the largest prime induced subgraph of G.

* mw = max of CCs

Chromatic number(mw)

Proposition

Chromatic number admits no polynomial compression in parameter mw, unless coNP $\subseteq$ NP$_{/poly}$.

AND cross composition

A problem $L \subseteq \Sigma^*$ AND-cross-composes into parameterized problem $Q \subseteq (\Sigma^* \times N)$ if, given t instances $x_1, \ldots, x_t$ of the same size n, one can output in polytime in tn an instance (I, k) of Q such that:

- $(I, k) \in Q$ if and only if $x_i \in Q$ for each $i = 1, \ldots, t$
- $k \leq n + \log t$

Theorem [Bodlaender, Jansen, Kratsch, 2014]

If an NP-hard problem L AND-cross-composes into Q, then Q has no polynomial compression, unless coNP $\subseteq$ NP$_{/\text{poly}}$.

Definition 7 (and-cross-composition)

Let $L \subseteq \Sigma^*$ be a set and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. We say that L *and-cross-composes* into Q if there is a polynomial equivalence relation R and an algorithm which, given t strings $x_1, \ldots, x_t$ belonging to the same equivalence class of R, computes an instance $(y, k) \in \Sigma^* \times \mathbb{N}$ in time polynomial in $\sum_{i=1}^{t} |x_i|$ such that:

1. the instance (y, k) is yes for Q iff all instances x_i are yes for L;
2. k is bounded by a polynomial in $\max_{i=1}^{t} |x_i| + \log t$.

Theorem 8

Let L be an NP-hard problem under Karp reductions. If L and-cross-composes into a parameterized problem Q, then Q does not admit a PC unless NP $\subseteq$ coNP/poly.

Chromatic number(*mw*)

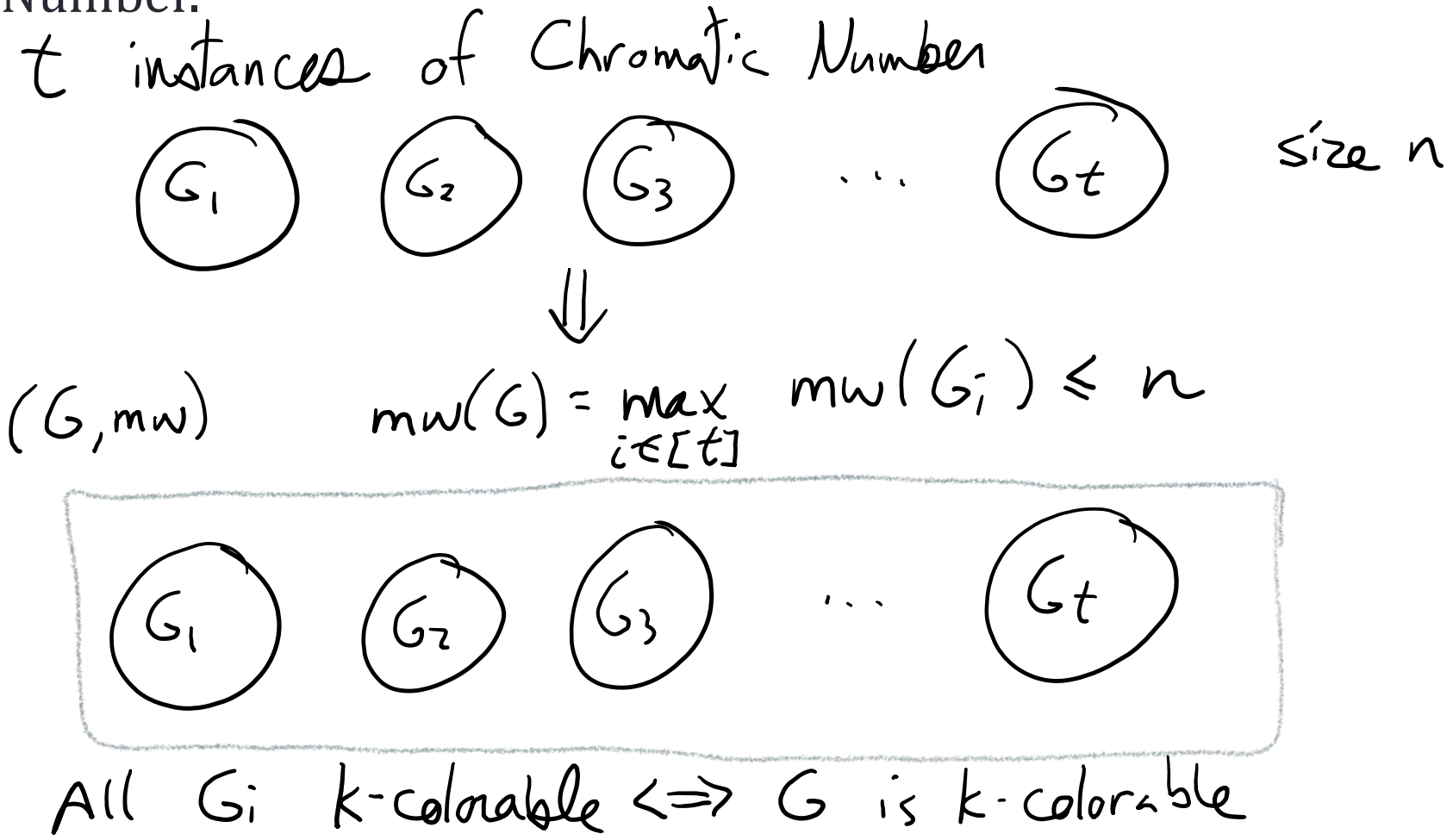
Proposition

Chromatic number admits no polynomial compression in parameter *mw*, unless coNP $\subseteq$ NP$_{/poly}$.

AND-cross-composition from (unparameterized) Chromatic Number.

Chromatic number(mw)

AND-cross-composition from (unparameterized) Chromatic Number.



Conclusion

- Many established techniques used in this paper
  - Reduction rules, MK-hardness reductions, compression, Turing kernels, cross-composition, nd encoding and MSO, ...
- Are there problems in structural parameterization that will require new techniques?
- Other structural parameters of interest?
  - feedback vertex set, treewidth, ...
- Meta-theorems for kernelization under structural parameters? (e.g. similar to the nd lemma)