A Polynomial Kernel for Diamond-Free Editing

CAO Yixin (操宜新)

Department of Computing, Hong Kong Polytechnic University

香港理工大學　電子計算學系

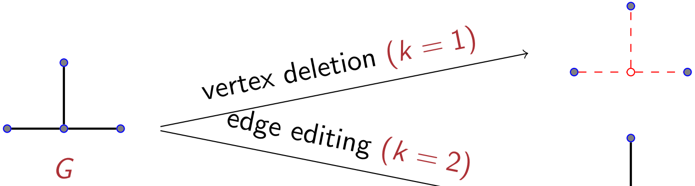
Workshop on Kernelization　　D-Day, 2019

Joint work with Ashutosh Rai, R. B. Sandeep, Junjie Ye

Graph modification problems

Graph modification problem for property P

Input: A graph G and an integer k.
Task: Can we apply $\leq k$ modifications to G to make it satisfy P?



P: being a cluster graph.

Three questions

1. Is it NP-complete?
2. Can it be solved in time $f(k) \cdot n^{O(1)}$; if yes, what is the (asymptotically) best f?
3. Does it have a polynomial kernel?

Theorem (Lewis and Yannakakis 80).
For a hereditary property, the vertex deletion problem is either NP-complete or trivial.

Such a general result for edge modification problems is unknown, and very unlikely.

(Complexity Status of Edge Modification Problems@FPT WIKI)

Three questions

1. Is it NP-complete?
2. Can it be solved in time $f(k) \cdot n^{O(1)}$; if yes, what is the (asymptotically) best f?
3. Does it have a polynomial kernel?

Theorem (Lewis and Yannakakis 80).
For a hereditary property, the vertex deletion problem is either NP-complete or trivial.

☹ Such a general result for edge modification problems is unknown, and very unlikely.

(Complexity Status of Edge Modification Problems@FPT WIKI)

1. Is it NP-complete?
2. Can it be solved in time $f(k) \cdot n^{O(1)}$; if yes, what is the (asymptotically) best f?
3. Does it have a polynomial kernel?

Theorem (Lewis and Yannakakis 80).
For a hereditary property, the vertex deletion problem is either NP-complete or trivial.

☹ Such a general result for edge modification problems is unknown, and very unlikely.

(Complexity Status of Edge Modification Problems@FPT WIKI)

Kernelization

An algorithm maps (I, k) into (I', k') in time polynomial in $|I| + k$ such that

- (I, k) is a yes-instance iff (I', k') is a yes-instance, and
- $|I'| + k' \leq g(k)$ for some computable function g.

The function g is the size of the kernel, and it is a polynomial kernel if g is polynomial.

H-free modification problems

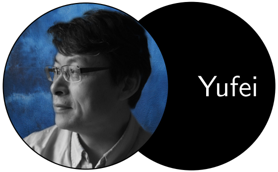
$|V(H)| \geq 2$

The "simplest" property is H-free—not containing H as an induced subgraph.
(For example, H is P_3 for cluster graphs.)

	vertex deletion	edge modification
NPC	Yes [Lewis & Yannakakis 81]	Dichotomy [Sandeep 16]
FPT	Yes [Cai 96]	Yes [Cai 96]
Poly kernel	Yes [Abu-Khzam 10]	?

no for someone

no for a lot

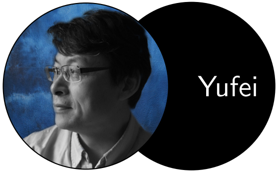


most no...

which has?

no for someone

no for a lot



most no...

which has?

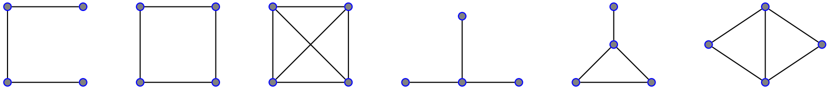
Drange & Mi. Pilipczuk:
"the existence of a polynomial kernel for any of H-Free Editing, H-Free Edge Deletion, or H-Free Completion problems is in fact a very rare phenomenon, and basically happens only for specific, constant-size graphs H."

What are known

H	completion	deletion	editing
$P_{\leq 4}$	yes	yes	yes
$P_{>4}$	no	no	no
$C_{\geq 4}$	no	no	no
K_ℓ	trivial	yes	yes
$K_{\geq 5} - e$	trivial	no	no
3-connected (≥ 2 non-edges)	no	no	no

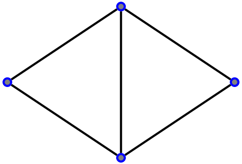
all yes when $|V(H)| = 2$ or 3.

Four-vertex graphs



(a) P_4 (b) C_4 (c) K_4 (d) claw (e) paw (f) diamond

H	completion	deletion	editing
P_4	$O(k^3)$ [Guillemot et al. 13]		
K_4	trivial	$O(k^4)$ [Cai 12]	
diamond	trivial	$O(k^3)$ [Sandeep & Sivadasan 15]	$O(k^8)$ [C et al. 2018]
claw	unknown		
paw	unknown		
C_4	no [Guillemot et al. 13]		



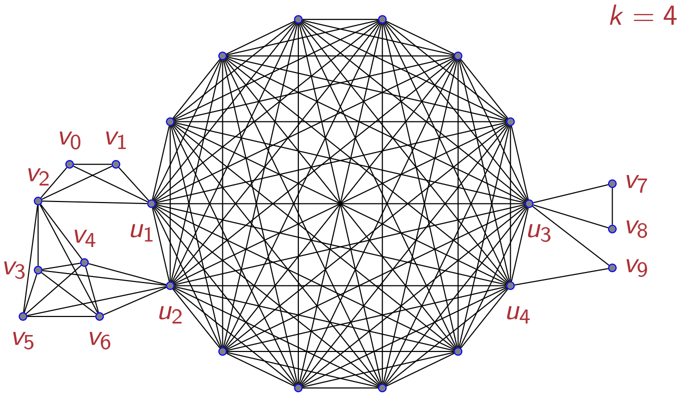
Diamond-free edge editing

Input: A graph G and an integer k.
Task: Can we edit (add/delete) $\leq k$ edges to make G diamond-free?

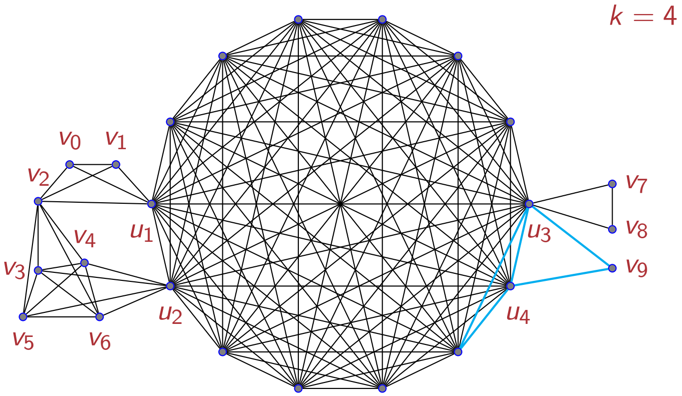
Main result: an $O(k^8)$-vertex kernel.

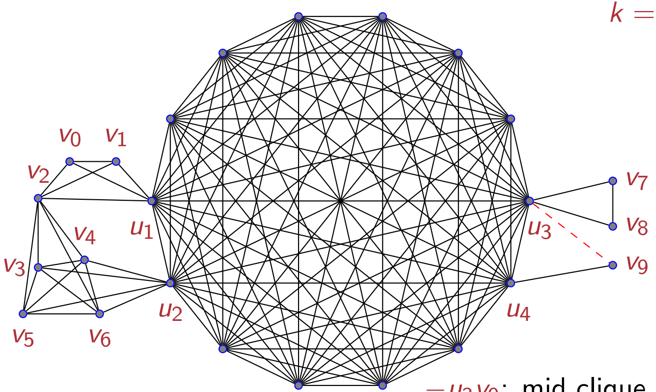
Maximal Cliques

An example



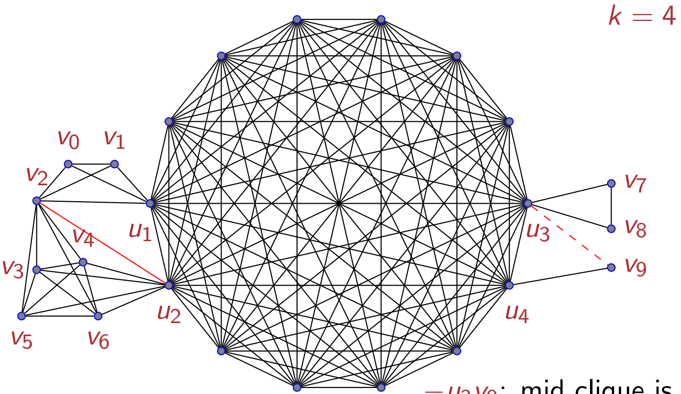
An example





$k = 4$

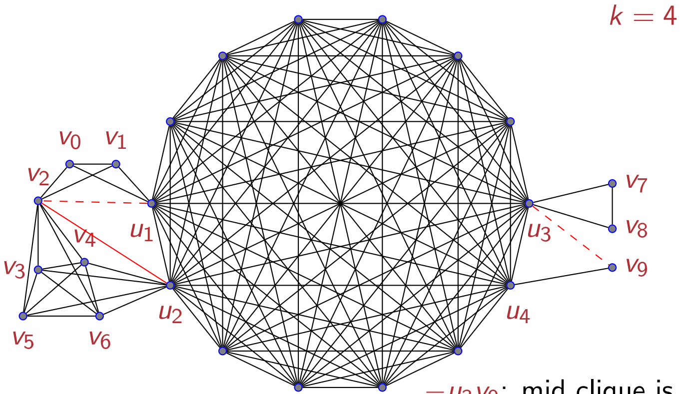
$-u_3 v_9$: mid clique is too big to fall.



$-u_3 v_9$: mid clique is too big to fall.

$+u_2 v_2$: shared by too many.

An example

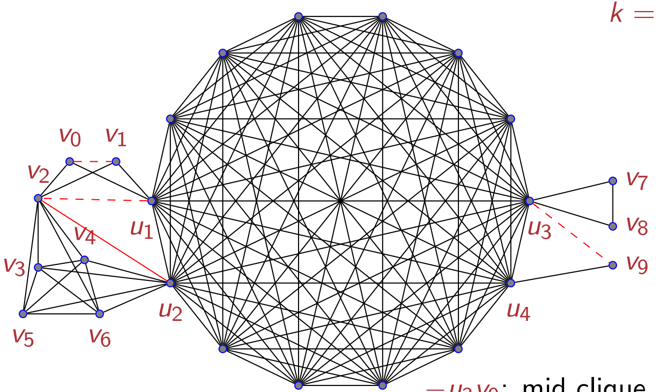


$-u_3 v_9$: mid clique is too big to fall.

$+u_2 v_2$: shared by too many.

$-u_1 v_2$: not in “original” diamonds.

An example



$k = 4$

$-u_3 v_9$: mid clique is too big to fall.

$+u_2 v_2$: shared by too many.

$-u_1 v_2$: not in "original" diamonds.

$-v_0 v_1$: not in "original" diamonds.

Maximal cliques

We are concerned with maximal cliques.

A maximal clique is of

type I if it shares ≥ 2 vertices with another maximal clique,

type II otherwise.

Maximal cliques

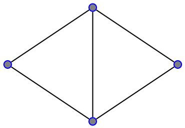
We are concerned with maximal cliques.

A maximal clique is of

type I if it shares ≥ 2 vertices with another maximal clique,

type II otherwise.

(simple) observation: diamond-free $\Leftrightarrow$ no type-I cliques.



Maximal cliques

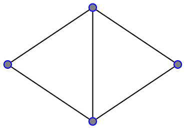
We are concerned with maximal cliques.

A maximal clique is of

type I if it shares ≥ 2 vertices with another maximal clique,

type II otherwise.

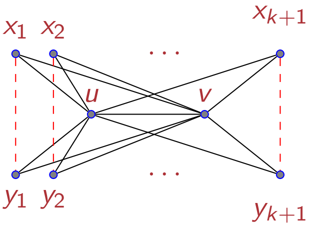
(simple) observation: diamond-free $\Leftrightarrow$ no type-I cliques.



A maximal clique K is *big* if $|K| \geq 3k + 2$, and *small* otherwise.

Basic rules

Rule 1(2).
If an edge (resp. non-edge) uv is the only edge (resp. non-edge) shared by $k+1$ diamonds, then delete (resp. add) uv, and decrease k by one.



We may assume throughout that Rules 1–2 have been exhaustively applied.

Properties: added edges

$E_{\pm}$ ($E_{+} \cup E_{-}$): a minimum solution to G; $G^{*} = G \triangle E_{\pm}$.

Proposition.
Big max cliques of G remain max cliques in G^{*}.

Properties: added edges

$E_\pm$ ($E_+ \cup E_-$): a minimum solution to G; $G^* = G \triangle E_\pm$.

Proposition.
Big max cliques of G remain max cliques in G^*.

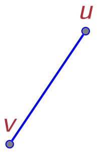
Lemma. Ends of E_+ are in small type-I cliques.

Properties: added edges

$E_{\pm}$ $(E_{+} \cup E_{-})$: a minimum solution to G; $G^{*} = G \triangle E_{\pm}$.

Proposition.
Big max cliques of G remain max cliques in G^{*}.

Lemma. Ends of E_{+} are in small type-I cliques.
Proof. Let $vu \in E_{+}$.

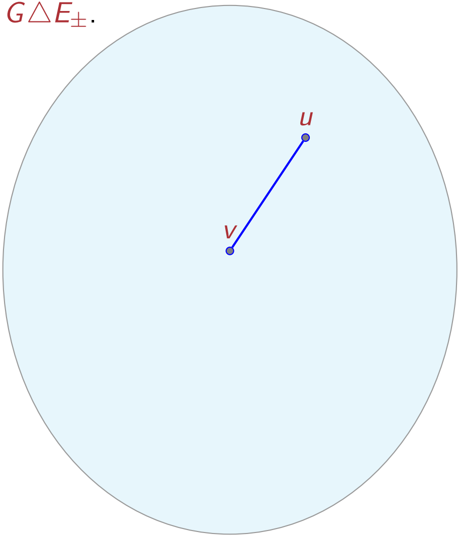


Properties: added edges

$E_\pm$ ($E_+ \cup E_-$): a minimum solution to G; $G^* = G \triangle E_\pm$.

Proposition.
Big max cliques of G remain max cliques in G^*.

Lemma. Ends of E_+ are in small type-I cliques.
Proof. Let $vu \in E_+$.
Find a maximal clique U of G^* containing u, v.



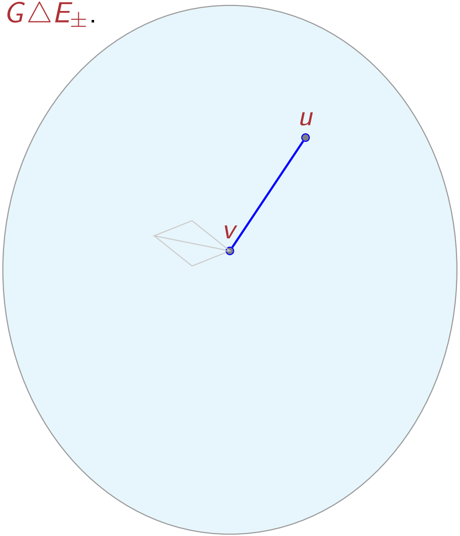
Properties: added edges

$E_\pm$ ($E_+ \cup E_-$): a minimum solution to G; $G^* = G \triangle E_\pm$.

Proposition.
Big max cliques of G remain max cliques in G^*.

Lemma. Ends of E_+ are in small type-I cliques.
Proof. Let $vu \in E_+$.
Find a maximal clique U of G^* containing u, v.

Claim: v is in a diamond in $G[U]$.

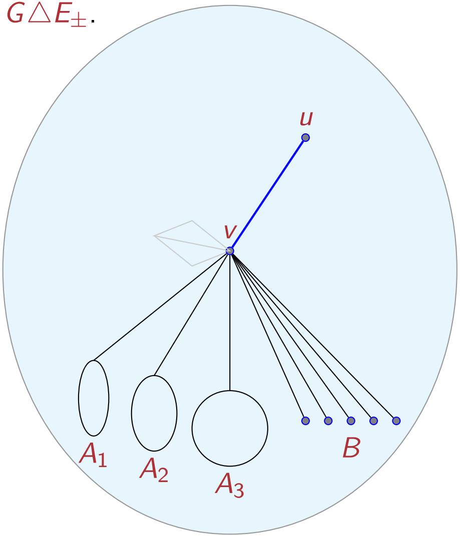


$E_\pm$ ($E_+ \cup E_-$): a minimum solution to G; $G^* = G \triangle E_\pm$.

Proposition.
Big max cliques of G remain max cliques in G^*.

Lemma. Ends of E_+ are in small type-I cliques.
Proof. Let $vu \in E_+$.
Find a maximal clique U of G^* containing u, v.

Claim: v is in a diamond in $G[U]$.
Otherwise, $N(v) \cup U$ induces a cluster graph.



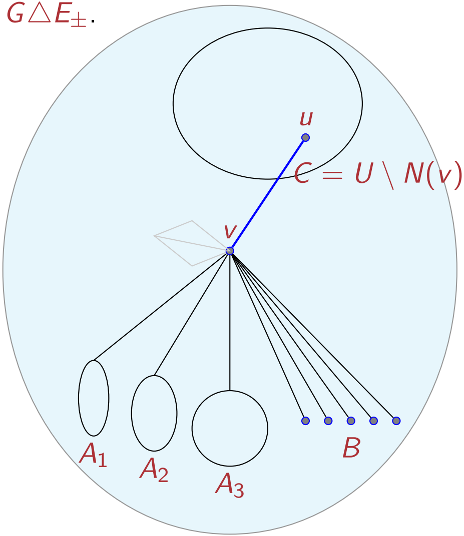
Properties: added edges

$E_\pm$ ($E_+ \cup E_-$): a minimum solution to G; $G^* = G \triangle E_\pm$.

Proposition.
Big max cliques of G remain max cliques in G^*.

Lemma. Ends of E_+ are in small type-I cliques.
Proof. Let $vu \in E_+$.
Find a maximal clique U of G^* containing u, v.

Claim: v is in a diamond in $G[U]$.
Otherwise, $N(v) \cup U$ induces a cluster graph.
We can find a smaller solution. □



Properties: added edges

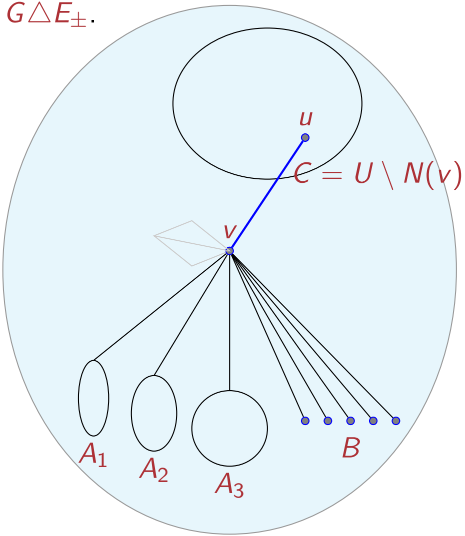
$E_\pm$ ($E_+ \cup E_-$): a minimum solution to G; $G^* = G \triangle E_\pm$.

Proposition.
Big max cliques of G remain max cliques in G^*.

Lemma. Ends of E_+ are in small type-I cliques.
Proof. Let $vu \in E_+$.
Find a maximal clique U of G^* containing u, v.

Claim: v is in a diamond in $G[U]$.
Otherwise, $N(v) \cup U$ induces a cluster graph.
We can find a smaller solution. □

A max type-I clique K of G containing v, x, y.



Properties: added edges

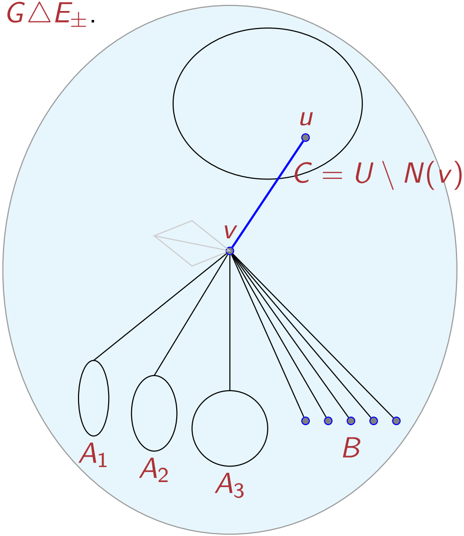
$E_\pm$ $(E_+ \cup E_-)$: a minimum solution to G; $G^* = G \triangle E_\pm$.

Proposition.
Big max cliques of G remain max cliques in G^*.

Lemma. Ends of E_+ are in small type-I cliques.
Proof. Let $vu \in E_+$.
Find a maximal clique U of G^* containing u, v.

Claim: v is in a diamond in $G[U]$.
Otherwise, $N(v) \cup U$ induces a cluster graph.
We can find a smaller solution. □

A max type-I clique K of G containing v, x, y.
$|K \cap U| > 2$, hence not a max clique of G^*.



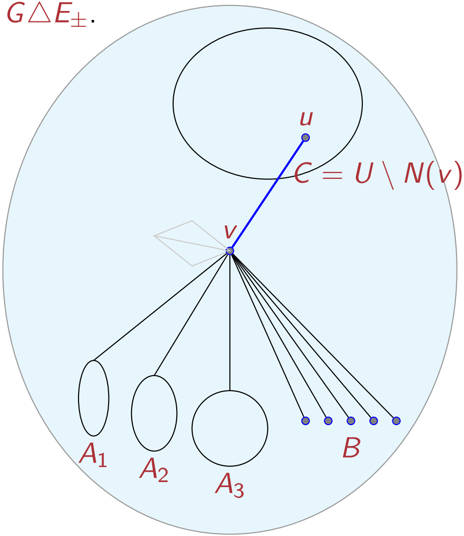
$E_{\pm}$ $(E_{+} \cup E_{-})$: a minimum solution to G; $G^{*} = G \triangle E_{\pm}$.

Proposition.
Big max cliques of G remain max cliques in G^{*}.

Lemma. Ends of E_{+} are in small type-I cliques.
Proof. Let $vu \in E_{+}$.
Find a maximal clique U of G^{*} containing u, v.

Claim: v is in a diamond in $G[U]$.
Otherwise, $N(v) \cup U$ induces a cluster graph.
We can find a smaller solution. □

A max type-I clique K of G containing v, x, y.
$|K \cap U| > 2$, hence not a max clique of G^{*}.
By the proposition above, K is not big. □



Properties: deleted edges

Lemma.
Let $uv \in E_-$. If a max clique K contains u, v, then K is small.
Moreover, if K is of type II, then K intersects some small type-I clique.

Vulnerable vertices

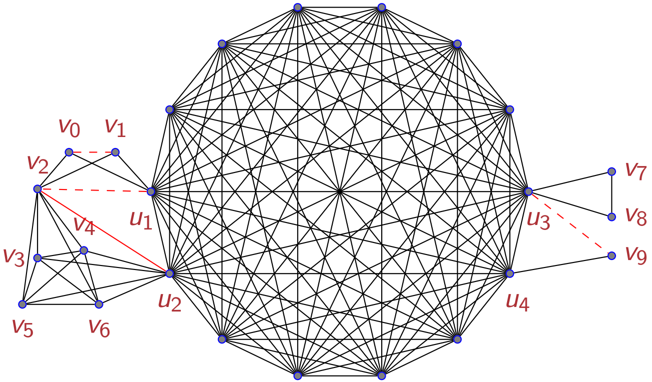
A vertex v is *vulnerable* if v is in

- some small type-I clique; or
- a type-II clique that intersects some small type-I clique.

Vulnerable vertices

A vertex v is *vulnerable* if v is in

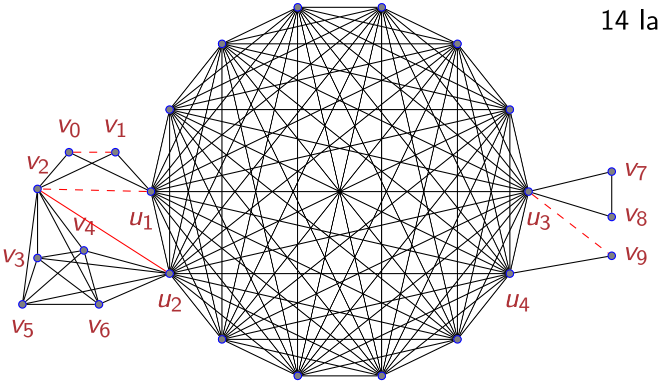
- some small type-I clique; or
- a type-II clique that intersects some small type-I clique.



Vulnerable vertices

A vertex v is *vulnerable* if v is in

- some small type-I clique; or
- a type-II clique that intersects some small type-I clique.

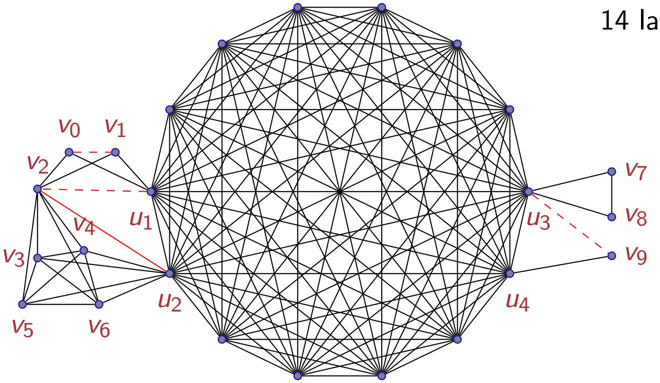


14 labeled vertices are vulnerable.
Others are *protected*.

Vulnerable vertices

A vertex v is *vulnerable* if v is in

- some small type-I clique; or
- a type-II clique that intersects some small type-I clique.



14 labeled vertices are vulnerable.
Others are *protected*.

Lemma. A minimum solution touches only vulnerable vertices.

Main rules

Lemma.
In a yes-instance, at most $18k^3 + 2k$ vertices are in small type-I cliques.

Big type-I cliques

Lemma.
A yes-instance has at most $6k^2$ big type-I cliques.

Rule 3.
If a big type-I clique contains a "private" protected vertex, delete it.
Lemma.
In a yes-instance, each big type-I clique has $O(k^3)$ vertices if Rule 3 is not applicable.

vertices in all type-I cliques: $O(k^3 + k^5) = O(k^5)$.

Big type-I cliques

Lemma.
A yes-instance has at most $6k^2$ big type-I cliques.

Rule 3.
If a big type-I clique contains a "private" protected vertex, delete it.
Lemma.
In a yes-instance, each big type-I clique has $O(k^3)$ vertices if Rule 3 is not applicable.

vertices in all type-I cliques: $O(k^3 + k^5) = O(k^5)$.

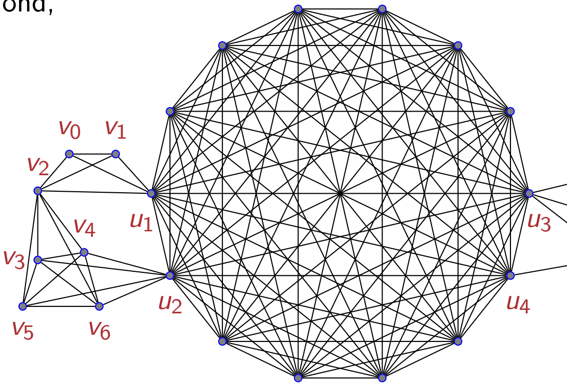
Type-II cliques

Let $T(G)$ denote the vertices that occur *only* in cliques of type II.

Rule 4.
If there is a protected vertex x contained in *only* in type-II cliques, delete it.

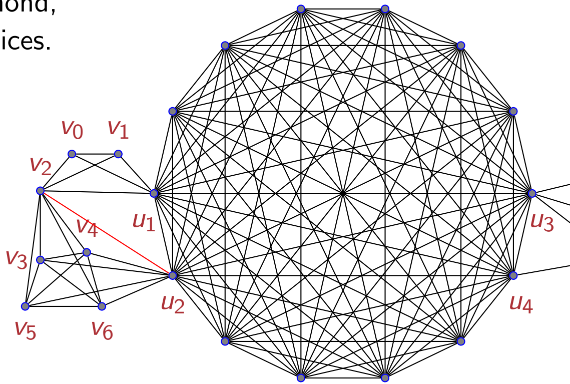
The last reduction rule

An edge in a type-II clique K is not in a diamond,



The last reduction rule

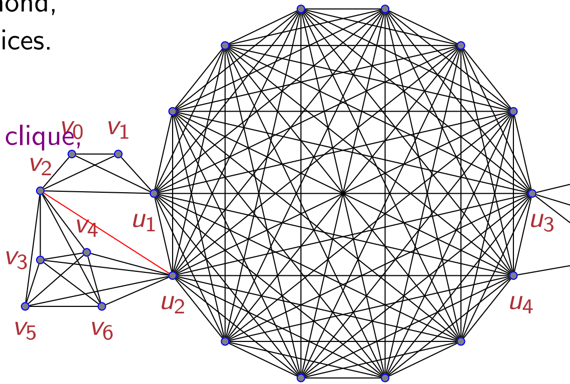
An edge in a type-II clique K is not in a diamond, unless edges added between K and other vertices.



The last reduction rule

An edge in a type-II clique K is not in a diamond, unless edges added between K and other vertices.

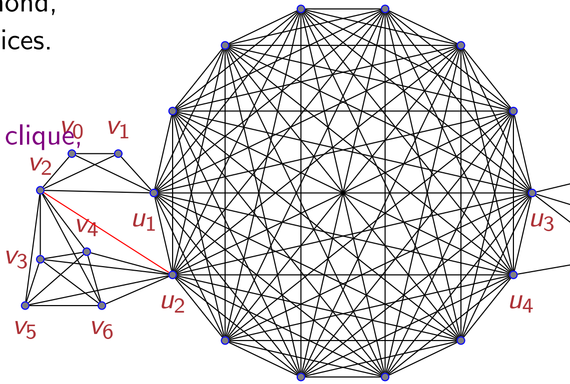
To maintain the immutability of a big type-II clique, it suffices to keep $k + 3$ of its vertices.



The last reduction rule

An edge in a type-II clique K is not in a diamond, unless edges added between K and other vertices.

To maintain the immutability of a big type-II clique, it suffices to keep $k+3$ of its vertices.



Rule 5. (Informal)

For each pair u, v in small type-I cliques

1. mark $k+1$ common neighbors of u, v;
2. for each marked neighbor w of u, mark $k+1$ common neighbors of u, w.

Delete all unmarked vertices x in $T(G)$.

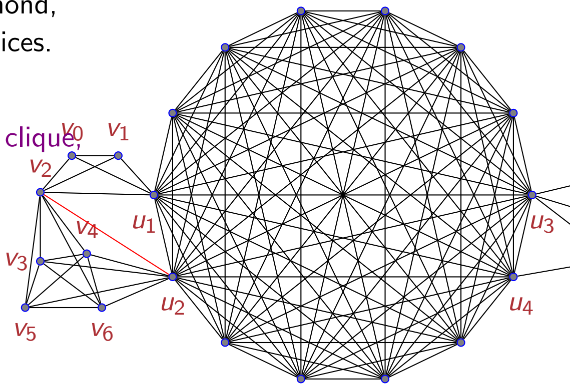
An edge in a type-II clique K is not in a diamond, unless edges added between K and other vertices.

To maintain the immutability of a big type-II clique, it suffices to keep $k+3$ of its vertices.

Rule 5. (Informal)
For each pair u, v in small type-I cliques

1. mark $k+1$ common neighbors of u, v;
2. for each marked neighbor w of u, mark $k+1$ common neighbors of u, w.

Delete all unmarked vertices x in $T(G)$.



In total, $O(k^8)$ vertices marked.

Putting together

$$O(k^3) + O(k^5) + O(k^8) = O(k^8).$$

The implementation

We do *not* need to enumerate the maximal cliques (exponential number of them).

It is sufficient to find

- all vertices in a small type-I clique,
- all vertices in a type-I clique, and
- all vulnerable vertices.

Key observation: we can enumerate cross edges of all diamonds, from which we can identify all vertices and edges in type-I cliques.

The implementation

We do *not* need to enumerate the maximal cliques (exponential number of them).

It is sufficient to find

- all vertices in a small type-I clique,
- all vertices in a type-I clique, and
- all vulnerable vertices.

Key observation: we can enumerate cross edges of all diamonds,
from which we can identify all vertices and edges in type-I cliques.

Diamond-free edge deletion

Diamond-free edge deletion

Input: A graph G and an integer k.
Task: Can we delete $\leq k$ edges to make G diamond-free?

Rule 2. If there is a vertex not in any small maximal clique, delete it.
Rule 3. Delete all edges and vertices not in any type-I clique.

Lemma.
The diamond-free edge deletion problem has a kernel of $O(k^3)$ vertices.

Michał: claw is difficult.

Can we take paw first (apparently attackable)?

Thanks!

yixin.cao@polyu.edu.hk