ADVANCED MEASURES

Hadas Shachnai

Recent Advances in Parameterized Complexity
Tel Aviv, December 4, 2017

OUTLINE

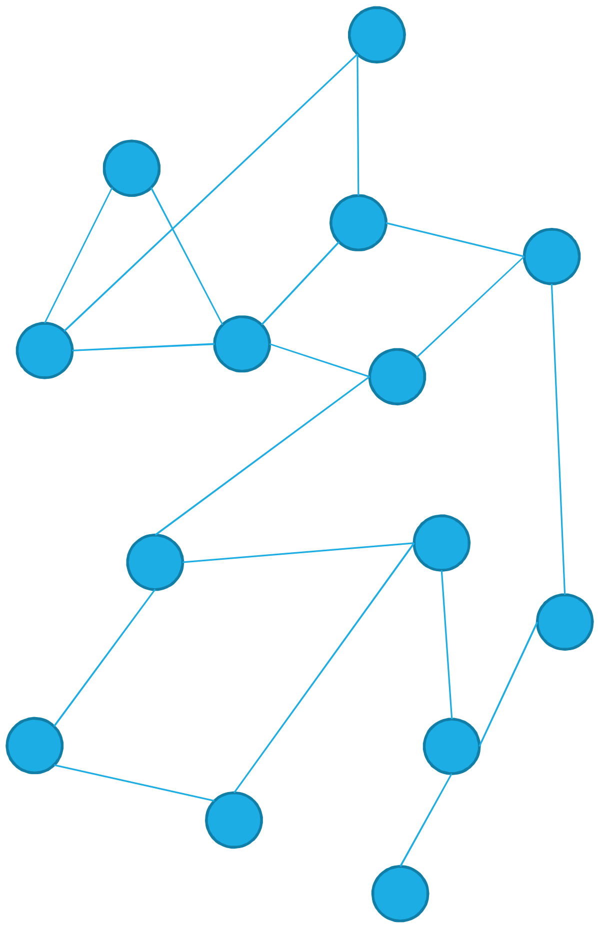
- Above Guarantee Parameterization
- Greedy Localization
- Using Matroid Theory in the Design of Parameterized Algorithms

ABOVE GUARANTEE PARAMETERIZATION

MAX CUT

G = (V,E)

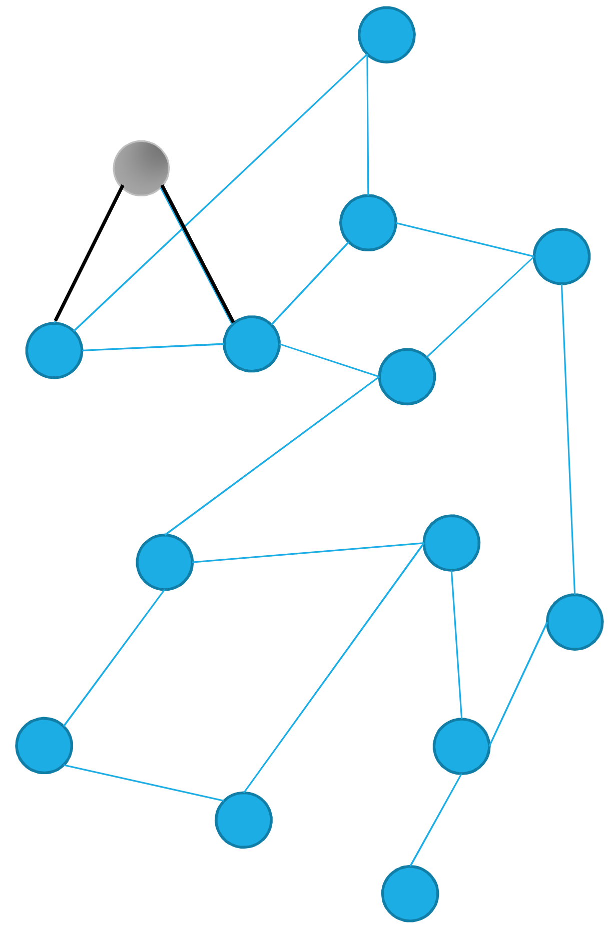
Find a subset $S \subseteq V$ such that $\left|E\left(S, \bar{S}\right)\right|$ is maximized.



MAX CUT

G = (V,E)

Find a subset $S \subseteq V$ such that $\left|E\left(S, \bar{S}\right)\right|$ is maximized.

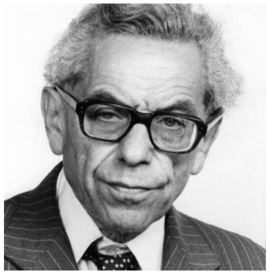


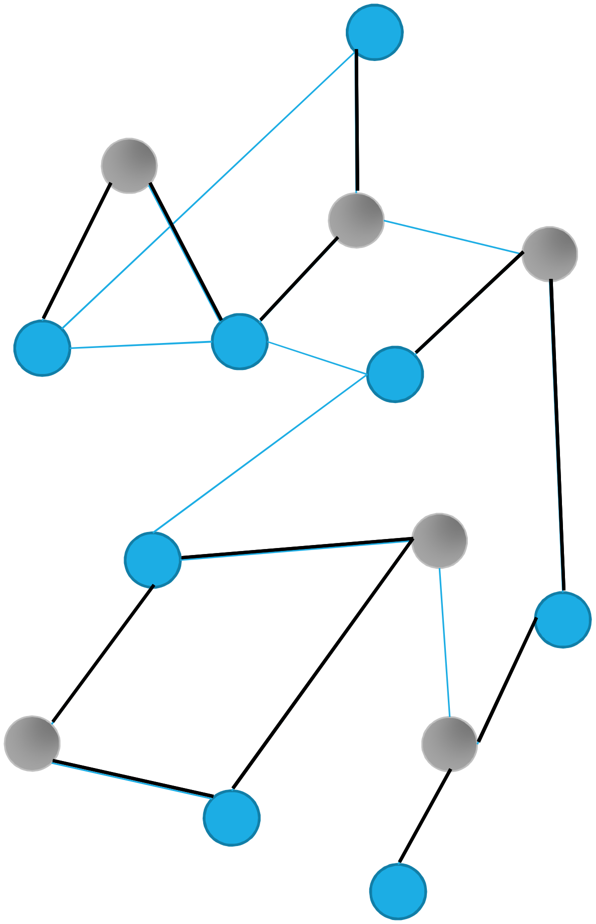
MAX CUT

G = (V,E)

Find a subset $S \subseteq V$ such that $\left|E\left(S,\bar{S}\right)\right|$ is maximized.

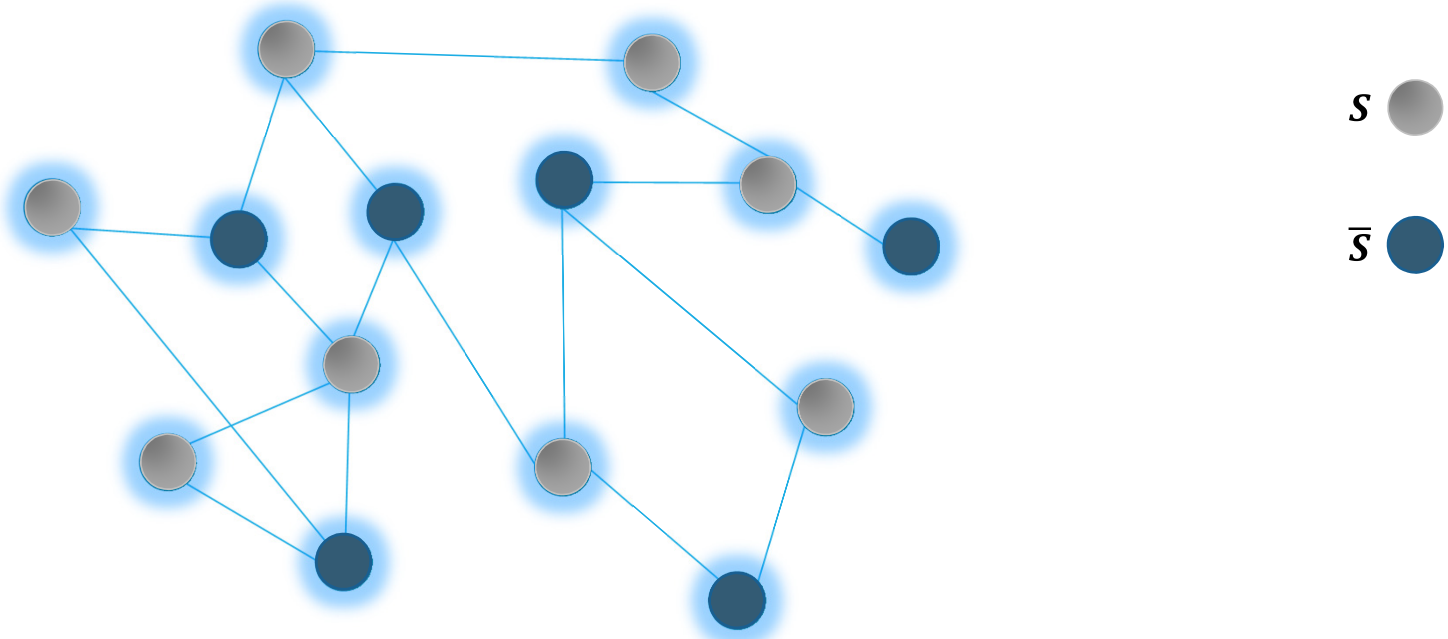
Any graph G with m=|E| edges has a cut of size $\frac{m}{2}$ (Erdős, 1965).





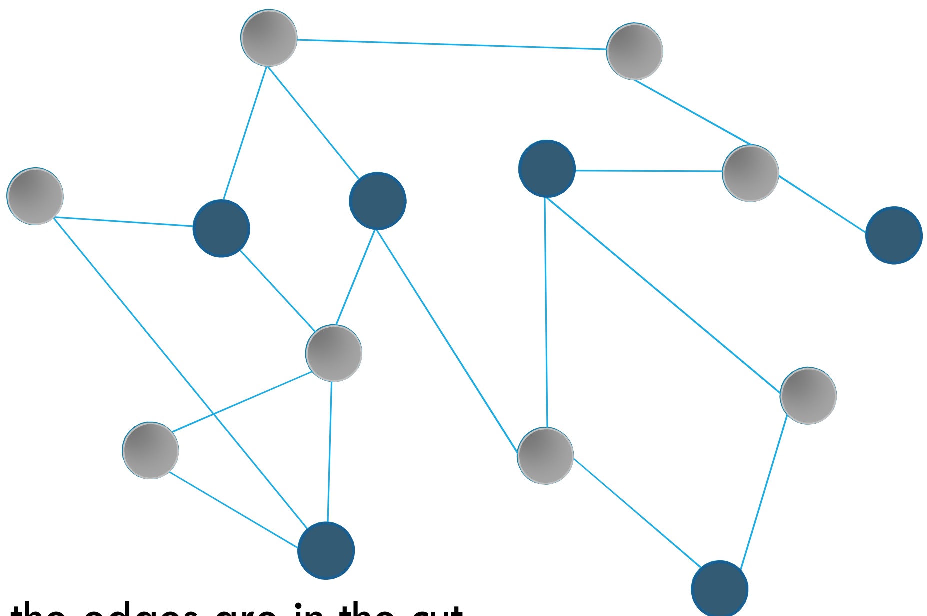
LOWER BOUND ON MAX CUT SIZE

Order the vertices arbitrarily. In iteration i, the i-th vertex makes a greedy decision,$1 \leq i \leq n$



LOWER BOUND ON MAX CUT SIZE

Order the vertices arbitrarily. In iteration i, the i-th vertex makes a greedy decision, $1 \leq i \leq n$



At least ½ of the edges are in the cut

MAX CUT: "STANDARD" PARAMETERIZATION

Max Cut (MC)
Instance: An undirected graph G and a positive integer k
Parameter: k
Question: Does G have a cut of size at least k?

- MC has an $O^*(1.2418^k)$ FPT algorithm.
- But, we have a guaranteed cut size of $\lceil \frac{m}{2} \rceil$.

If k$\leq \lceil \frac{m}{2} \rceil$ the answer is always "Yes", and for large values of k - the FPT algorithm may not be efficient.

MAX CUT: "STANDARD" PARAMETERIZATION

Max Cut (MC)
Instance: An undirected graph G and a positive integer k
Parameter: k
Question: Does G have a cut of size at least k?

- Refine the question: How much larger than the guaranteed value is the maximum value?

MAX CUT: ABOVE GUARANTEE PARAMETERIZATION

Above Guarantee Max Cut (AGMC)

Instance: An undirected graph G=(V,E), |E|=m, and a positive integer k

Parameter: k

Question: Does G have a cut of size at least k+ $\left\lceil \frac{m}{2} \right\rceil$ *?*

- Assume w.l.o.g that G has no isolated vertices and no self loops.
- Use a tighter guarantee (Edwards-Erdős): Every connected graph G=(V, E) where |V|=n and |E|=m with no self loops has a cut of size at least $\frac{m}{2} + \frac{n-1}{4}$.

⇒ Any graph G with c connected components has a cut of size $\frac{m}{2} + \frac{n-c}{4}$

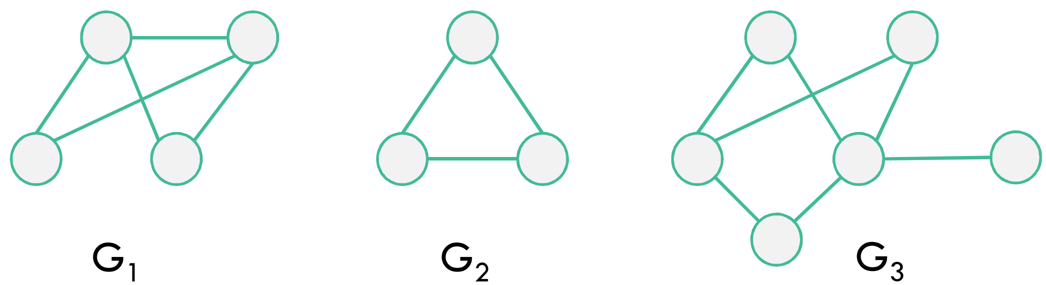
MAX CUT: ABOVE GUARANTEE PARAMETERIZATION

FPT Algorithm for AGMC

1. Find the connected components $G_{1,}G_{2,}\ldots, G_c$ of G. If $k \leq \frac{n-c}{4}$ output Yes and stop.
2. For i=1, ..., c find the max cut size, s_i, in G_i.
3. If $\sum_{i=1}^{c} s_i \geq \left\lceil \frac{m}{2} \right\rceil + k$ then output Yes; else, output No.

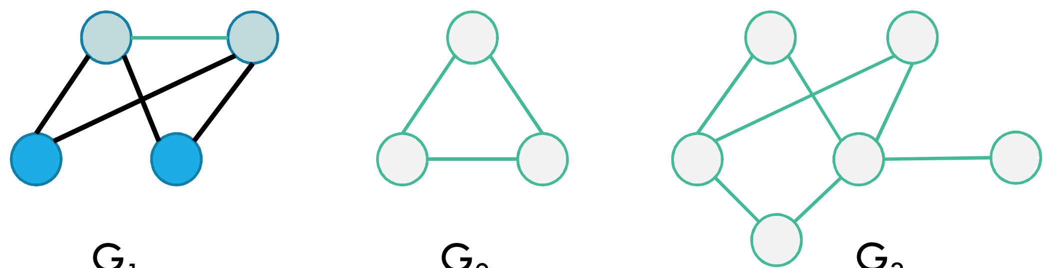
MAX CUT: ABOVE GUARANTEE PARAMETERIZATION

Example: k=3.
c=3, n=13, m=14,
k > $\frac{n-c}{4}$



MAX CUT: ABOVE GUARANTEE PARAMETERIZATION

Example: k=3.
c=3, n=13, m=14,
k > $\frac{n-c}{4}$

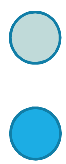


G_1 G_2 G_3

$S_1=4$

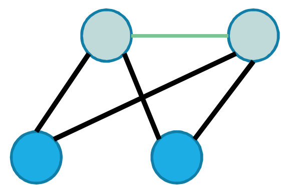
S

$\overline{S}$



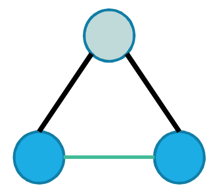
MAX CUT: ABOVE GUARANTEE PARAMETERIZATION

Example: k=3.
c=3, n=13, m=14,
k > $\frac{n-c}{4}$



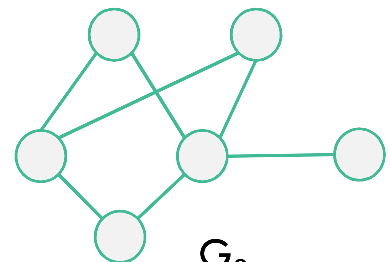
G_1

S_1=4



G_2

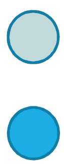
S_2=2



G_3

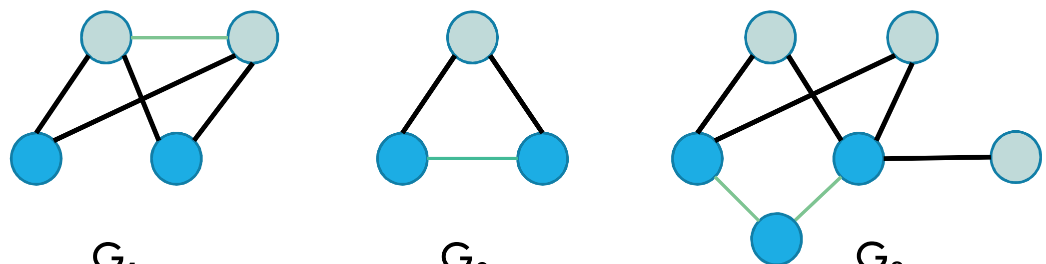
S

$\overline{S}$



MAX CUT: ABOVE GUARANTEE PARAMETERIZATION

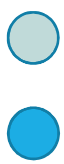
Example: k=3.
c=3, n=13, m=14,
k $> \frac{n-c}{4}$



G_1 $s_1=4$

G_2 $s_2=2$

G_3 $s_3=5$

S 

$\overline{\text{S}}$

$\sum_{i=1}^{c} s_i = 11, \quad \left\lceil \frac{m}{2} \right\rceil + k = 10$

$\sum_{i=1}^{c} s_i \geq \left\lceil \frac{m}{2} \right\rceil + k$ ➔ YES

MAX CUT: ABOVE GUARANTEE PARAMETERIZATION

FPT Algorithm for AGMC

1. Find the connected components $G_1, G_2, \ldots, G_c$ of G. If k $\leq \frac{n-c}{4}$ output Yes and stop.
2. For i=1, …, c find the max cut size, s_i, in G_i.
3. If $\sum_{i=1}^{c} s_i \geq \left\lceil \frac{m}{2} \right\rceil + $ k then output Yes, else output No.

- Running time is $O(n^3 + m \cdot 2^{4k})$

Note: In Step. 2 it holds that $n\text{-}c < 4k$. It follows that, for all $1 \leq i \leq c,\ n_i \leq n - (c-1) \leq 4k$.

- If $k = O(\log mn)$, running time is polynomial in m and n.

ABOVE GUARANTEE VERTEX COVER

Let $\mu(G)$ be the size of a maximum matching in G.

Above Guarantee Vertex Cover (AGVC)

Instance: An undirected graph G=(V,E), and a positive integers k, $\mu(G)$.

Parameter: k $- \mu(G)$

Question: Does G have a vertex cover of size at most k?

- An $O^*(4^{k-\mu(G)})$ algorithm follows from a reduction to Node Multiway Cut [Cygan et al., 2011]
- Better running time can be obtained by considering the linear programming (LP) formulation.

ABOVE GUARANTEE VERTEX COVER

ILP for Vertex Cover

Instance: A graph G=(V,E), and a positive integer k

Feasible Solution: A function x:V→{0,1} satisfying x(u)+x(v) ≥ 1 *for any edge (u,v)∈ E*

Goal: Minimize w(x)= $\sum_{u \in V} x(u)$ *over all feasible solutions x.*

- In the LP relaxation x(v) ≥ 0 for all v ∈ *V*.
- Let vc^*(G) be the minimum value for the LP.

Vertex Cover Above LP

Instance: An undirected graph G, and positive integers k, $\lceil vc^*(G) \rceil$

Parameter: k $- \lceil vc^*(G) \rceil$

Question: Does G have a vertex cover of size at most k?

ABOVE GUARANTEE VERTEX COVER

Vertex Cover Above LP
Instance: *An undirected graph G, and positive integers k,* $\lceil vc^*(\mathrm{G}) \rceil$
Parameter: $k - \lceil vc^*(\mathrm{G}) \rceil$
Question: Does G have a vertex cover of size at most k?

- VC above LP can be solved in $O^*(2.6181^{k-vc^*(G)})$ [Narayanaswamy et al., 2012]
- By weak duality for linear programs, $vc^*(\mathrm{G}) \geq \mu(G)$; thus, $k - vc^*(\mathrm{G}) \leq k - \mu(G)$.

Any algorithm for VC Above LP is also an algorithm for AGVC.

THE ZOO AROUND AGVC

- Faster algorithms for AGVC yield faster algorithms for many other problems:

✓ Almost 2-SAT

✓ Odd Cycle Transversal

✓ Edge Bipartizaton

✓ Split Vertex Deletion

✓ …

THE ZOO AROUND AGVC: EXAMPLE

Odd Cycle Transversal (OCT)
Instance: An undirected graph G and a positive integer k
Parameter: k
Question: Does G have an odd cycle transversal of size at most k?

Theorem: Odd Cycle Transversal can be solved in time $O^*(4^k)$

Proof: Given the input graph G, construct the graph $\tilde{G}$.

- Let G_1 =(V_1, E_1) and G_2 =(V_2, E_2) be two copies of G. Then,

V($\tilde{G}$) = $V_1 \cup V_2$ and E($\tilde{G}$)= $E_1 \cup E_2 \cup \{u_1u_2 : u \in$ V(G)}

- Let M={u_1u_2 : $u \in$ V(G)} be a perfect matching in $\tilde{G}$, where |M|=n.

THE ZOO AROUND AGVC: EXAMPLE

Theorem: Odd Cycle Transversal can be solved in time $O^*(4^k)$

Proof: Given the input graph G, construct the graph $\tilde{G}$.

- Let G_1 =(V_1, E_1) and G_2 =(V_2, E_2) be two copies of G. Then,

V($\tilde{G}$) = $V_1 \cup V_2$ and E($\tilde{G}$)= $E_1 \cup E_2 \cup \{u_1 u_2 : u \in$ V(G)$\}$

- By the construction, $\mu(\tilde{G})$=n.

Lemma: G has OCT of size at most k ⟺ $\tilde{G}$ has a VC of size at most k+n.

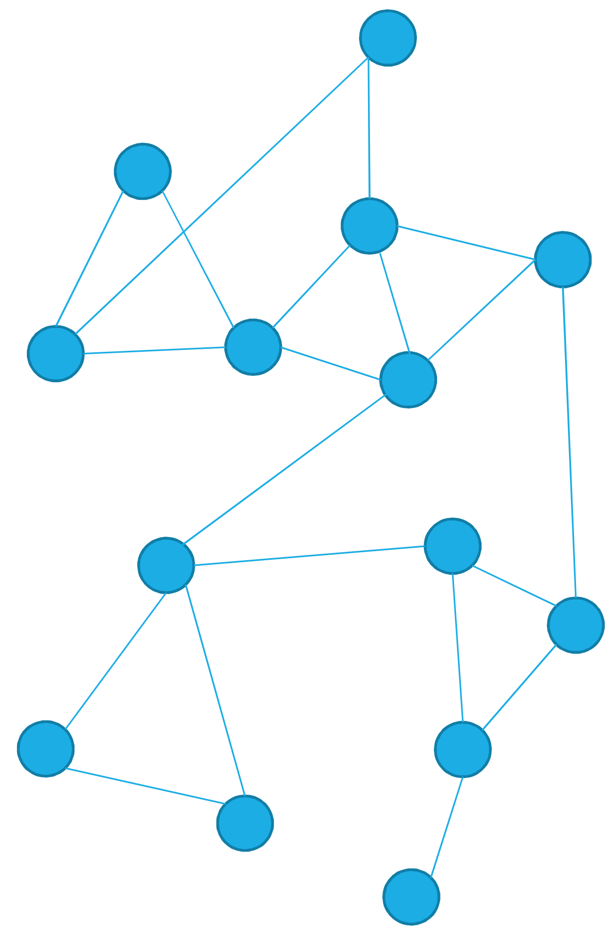
Solve AGVC on $\tilde{G}$ with parameter k+n−$\mu(\tilde{G})$=k, in time $O^*(4^k)$. If Yes, use the Lemma to obtain an OCT of size at most k in G. Otherwise (by the Lemma), G has no OCT of size at most k. ■

GREEDY LOCALIZATION

TRIANGLE PACKING

G = (V,E)

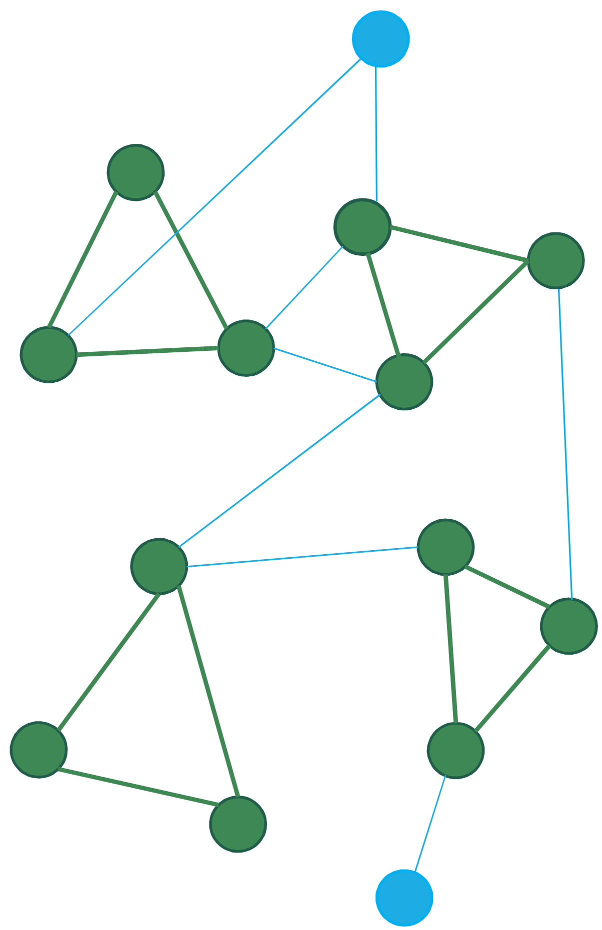
Find a maximum size set of vertex disjoint triangles.



TRIANGLE PACKING

$G = (V,E)$

Find a maximum size set of vertex disjoint triangles.



PARAMETERIZING TRIANGLE PACKING

k-Packing of Triangles
Instance: An undirected graph G
Parameter: A positive integer k
Question: Does G have k disjoint copies of K_3*?*

- Use branching to recursively search for a solution.
- Can we enhance the branching?

Start off with a Greedy step!

K-PACKING OF TRIANGLES: GREEDY LOCALIZATION

k-Packing of Triangles
Instance: An undirected graph G
Parameter: A positive integer k
Question: Does G have k disjoint copies of K_3*?*

Step 1: Find a maximal set A of vertex-disjoint triangles in G

Observation: If there exists a solution for the problem, SOL (containing r $\geq$ k disjoint triangles), any triangle in SOL intersects a triangle in A (otherwise A is not maximal).

Guess the intersection of A and SOL

K-PACKING OF TRIANGLES: GREEDY LOCALIZATION

k-Packing of Triangles
Instance: An undirected graph G
Parameter: A positive integer k
Question: Does G have k vertex disjoint triangles?

Step 1: Find a maximal set A of vertex-disjoint triangles in G

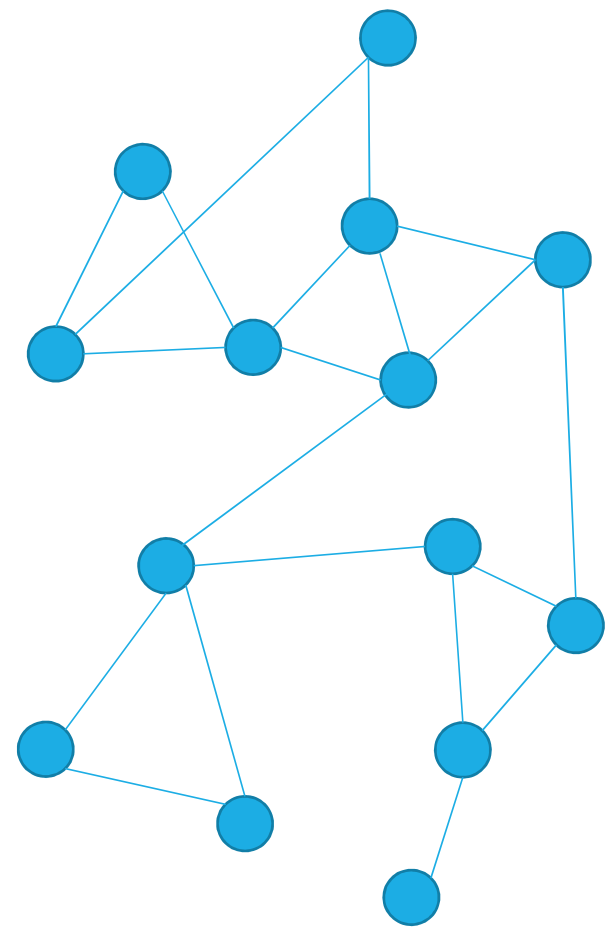
Step 2: If |A| $\geq$ k return Yes and stop; else, let G_A be the subgraph induced by A.

Step 3: If |V(G_A)|< k return No and stop; else, guess a subset S of k vertices in V(G_A).
Each vertex in v_i ∈S is a 'partial triangle' O_i in a solution SOL for G.

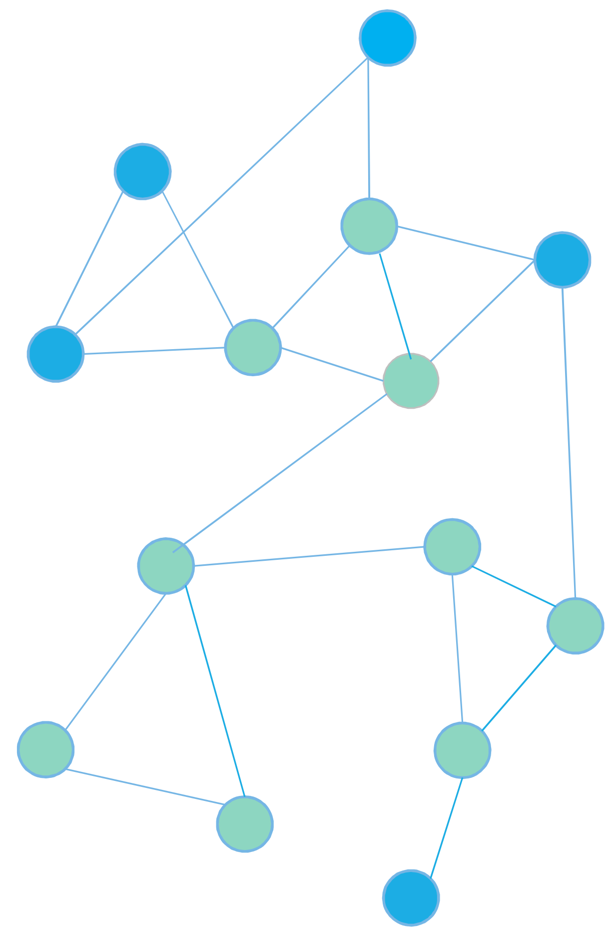
Step 4: Guess the extension of all partial triangles to the full ones.

K-PACKING OF TRIANGLES: GREEDY LOCALIZATION

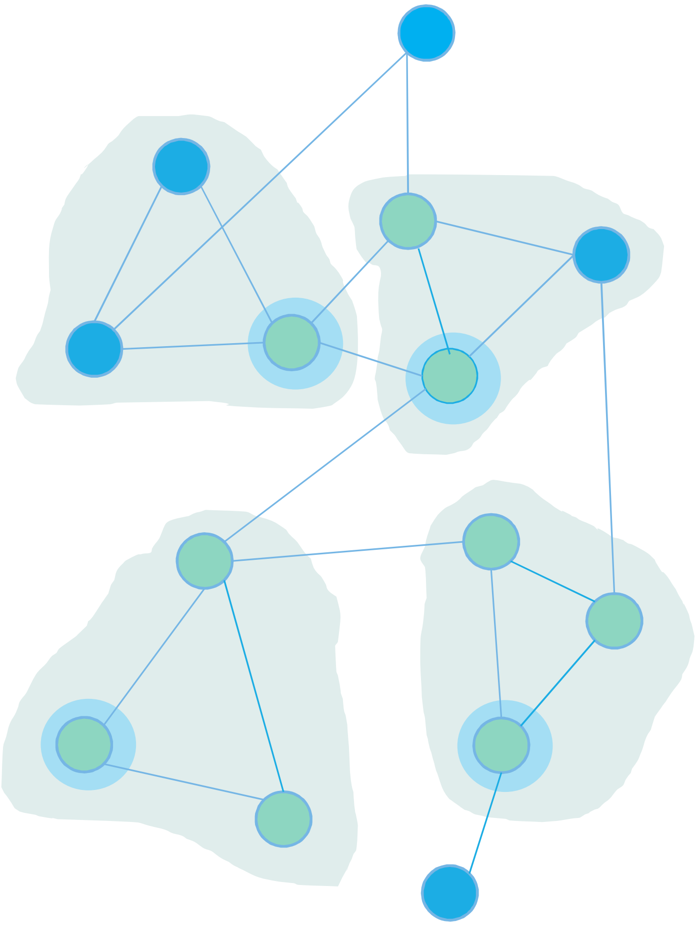
Example: k=4



K-PACKING OF TRIANGLES: GREEDY LOCALIZATION



K-PACKING OF TRIANGLES: GREEDY LOCALIZATION



K-PACKING OF TRIANGLES: GREEDY LOCALIZATION

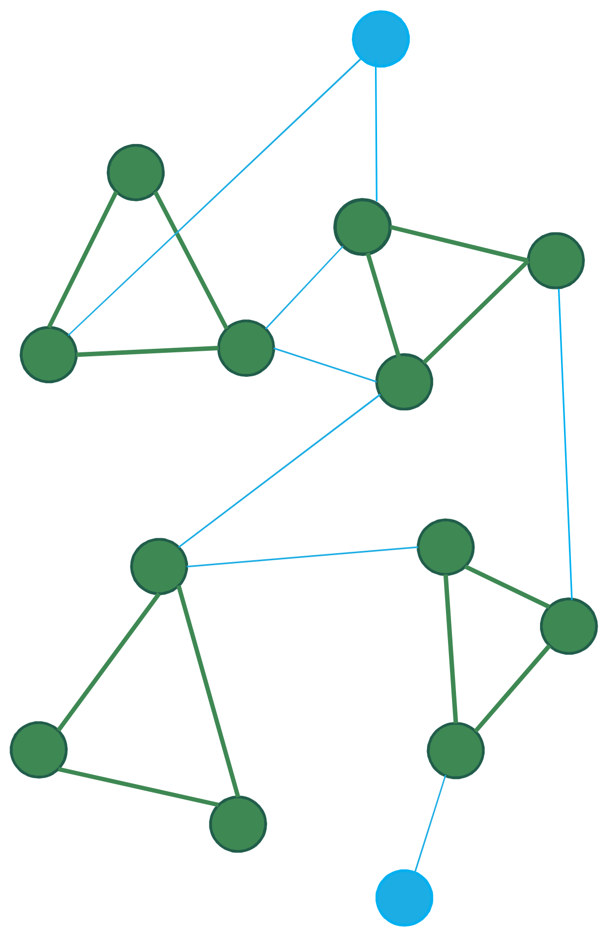
Running time:

- Guessing S requires $2^{O(k)}$ steps (since $|V(G_A)| \leq 3(k-1)$)
- Guessing the vertices that complete the partial triangles requires $k^{O(k)}$ steps.

Note: Given the 'correct' k vertices in S, if a partial triangle O_i cannot be completed, we need to 'guess' a vertex among those that were added to $O_1, \ldots, O_{i-1}$, and add this vertex to O_i. There are O(k) such vertices.

We need to 'guess' such a vertex, to be added to a partial triangle, at most 3k times, since the total number of vertices in the solution is 3k.

- Overall running time $O^*(k^{O(k)})$=O^* $(2^{O(k \log k)})$



GREEDY LOCALIZATION – GENERAL TECHNIQUE

- Let Π be a parameterized problem satisfying the following.

Property 1: Π can be formulated as a problem of finding k-pairwise disjoint objects in an input instance G, where an object is a subset of elements in the ground set V of G, whose size depends only on k.

Property 2: For any R⊆V and X⊆V, we can decide in FPT time if there exists

$S \subseteq V \setminus X$, such that $R \cup S$ is an object (S is an extension of the partial object R to a full object $R \cup S$ avoiding X).

Notation: For a set of (partial) objects B={B_1, B_2,…, B_k}, let V_B be the collection of ground elements contained in B.

GREEDY LOCALIZATION – GENERAL TECHNIQUE

Meta-algorithm Greedy Localization
Input: Instance G with ground set V, an integer k.
Output: Yes if G has k non-overlapping objects, otherwise No.

1. Compute an inclusion maximal non-overlapping set of objects A.
2. If A contains at least k objects then return Yes and stop.
3. If $|V_A| <$ k, return No and stop.
4. Guess $\{v_{1,} v_{2,} \dots, v_k\} \subseteq V_A$, and let O_i be a partial object on v_i.
   Branching(O $= \{O_1, O_2 \dots, O_k\}$)

- The subroutine Branching accepts as input a set of k objects $O = \{O_1, O_2 \dots, O_k\}$ and returns Yes if O can be extended to a set of full objects; else returns No.

GREEDY LOCALIZATION – GENERAL TECHNIQUE

Theorem: If a parameterized problem Π satisfies Properties 1 and 2 then Greedy Localization is an FPT algorithm for Π.

Proof: For correctness, the key observation is that if ∃ a solution for Π,
SOL={SOL_1, …, SOL_r}, where r$\geq k$, then any object SOL_i intersects an object A_j ∈ A (otherwise A is not maximal).

- We can guess k ground elements in V_A, each belongs to a distinct object in SOL.
- The subroutine Branching will find for each one element partial object O_j an extension to an element SOL_i.

GREEDY LOCALIZATION – GENERAL TECHNIQUE

Theorem: If a parameterized problem Π satisfies Properties 1 and 2 then Greedy Localization is an FPT algorithm for Π.

Proof: For running time, the initial guess of k elements requires $\binom{|V_A|}{k}$, which is a function of k.

- Guessing the extension of each element to an object in SOL can be done in FPT time, by Property 2, and since object sizes are bounded by a function of k.

Overall, we have FPT time.

■

GREEDY LOCALIZATION: SOME EXAMPLES

- The Technique was first used by [Chen et al., 2004] and [Jia et al., 2004] . The term was coined by [Dehne et al., 2004].
- Applied to solve:

✓ k-Packing of k-Cycles

✓ k-Set Packing

✓ k-Dimensional Matching

✓ Multi-layer Cluster Editing

✓ Cochromatic Number

✓ Disjoint Rectangle Stabbing

✓ …

MATROID THEORY AND FPT ALGORITHMS

MATROIDS: INTRODUCTION

Let U be a finite universe. A matroid $\mathcal{M}$ is a pair $\mathcal{M} = (U, \mathcal{F})$ satisfying:

1. $\mathcal{F}$ is a collection of subsets of U (called independent sets).
2. Hereditary: $B \in \mathcal{F}$ and $A \subseteq B$ imply $A \in \mathcal{F}$.
3. Exchange property: if $A \in \mathcal{F}$ and $B \in \mathcal{F}$ and $|A| < |B|$, then $\exists x \in B \setminus A$ such that $A \cup \{x\} \in \mathcal{F}$.

- A basis is a maximum size independent set.
- By the exchange property, all bases of a matroid $(U, \mathcal{F})$ have the same size (called the rank of the matroid).

Example:
$U=\{1,2,3\}$
$\mathcal{F}= \{\{1,2\} \{2,3\},\{1\},\{2\},\{3\},\emptyset\}.$

$\mathcal{M} = (U, \mathcal{F})$ is a matroid

WEIGHTED MATROIDS

- A matroid $\mathcal{M} = (U, \mathcal{F})$ is weighted if there is a weight function w: $U \rightarrow R^{+}$.
- For a subset $S \subseteq U$ let $w(S) = \sum_{u \in S} w(u)$
- The Matroid optimization problem: Find a basis S of minimum total weight

Matroid-Greedy Algorithm
1. $S \leftarrow \emptyset$
2. Sort the elements in U in non-decreasing order by weights, so that $w(u_1) \leq \cdots \leq w(u_n)$
3. *For i=1 to n do*
   if $S \cup \{u_i\} \in \mathcal{F}$ then $S \leftarrow S \cup \{u_i\}$
4. Output S

MATROID OPTIMIZATION PROBLEM: EXAMPLE

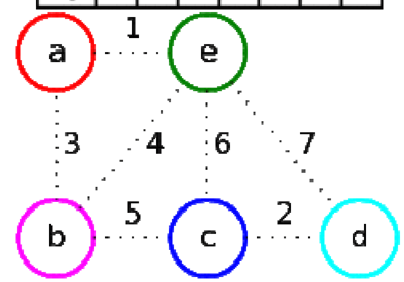
- Given a connected graph G=(V, E) with w: E $\longrightarrow R^+$, find a minimum spanning tree (MST) for G.

MST-Kruskal

1. T $\longleftarrow \emptyset$
2. Sort the edges in E in non-decreasing order by weights, so that $w(e_1) \leq \cdots \leq w(e_m)$
3. *For i=1 to m do*

   if $T \cup \{e_i\}$ does not contain a cycle then T$\longleftarrow T \cup \{e_i\}$
4. Output T

Edge	ab	ae	bc	be	cd	ed	ec
Weight	3	1	5	4	2	7	6



- Let U=E and $\mathcal{F}$={ E' | (V, E') is a forest}, then $\mathcal{M}$ = (U, $\mathcal{F}$) is a graphic matroid.

Kruskal's Algorithm is an application of Matroid-Greedy on $\mathcal{M}$!

THE MATROID PARITY PROBLEM

Matroid parity

Instance: A matroid $\mathcal{M}$ = (U, $\mathcal{F}$), $\mathcal{P} \subseteq \binom{U}{2}$ a collection of pairs of elements in U.

Parameter: A positive integer k

Question: Is there a subset S $\subseteq \mathcal{P}$ of at least k disjoint pairs, whose union is an independent set in $\mathcal{M}$?

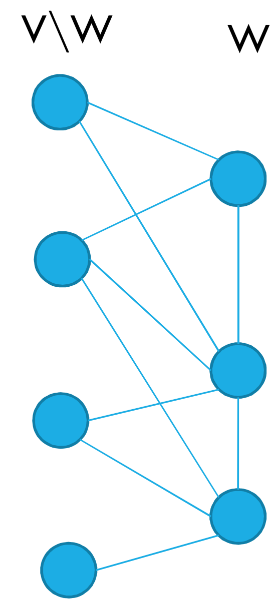
Theorem: Matroid Parity admits polynomial-time algorithm.

FEEDBACK VERTEX SET IN SUBCUBIC GRAPHS

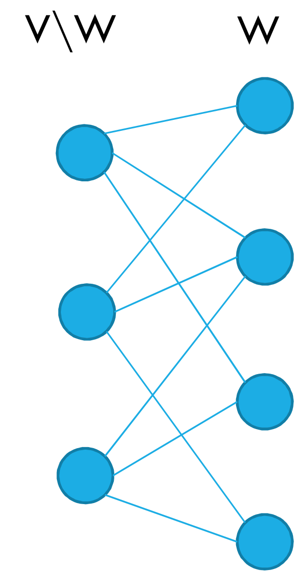
FVS in Subcubic Graph
Instance: A graph G=(V,E) of maximum degree $\Delta \leq 3$
Parameter: A positive integer k
Question: Is there a subset $S \subseteq V$ of at most k vertices, such that $G-S$ is acyclic?

Consider first a variant of FVS.

Special Disjoint FVS
Instance: A graph G=(V,E) and $W \subseteq V$ such that $G-W$ is independent, and each vertex in $V \setminus W$ has degree at most 3.
Parameter: A positive integer k
Question: Is *there a subset* $S \subseteq V \setminus W$ of at most k vertices, such that *$G-S$ is acyclic?*



FVS IN SUBCUBIC GRAPHS (CONT.)



Lemma: Special Disjoint FVS has a polynomial-time algorithm.

Proof (Sketch): Given an input graph of Special Disjoint FVS, use exhaustively reduction rules to obtain an instance that is a bipartite graph, G=($V\backslash W$, W, E), and every vertex in $V\backslash W$ has degree exactly 3.

➡ Use a reduction to the Matroid Parity problem.

- Let H=(W, E_H) be a clique with vertex set W, and $\mathcal{M}_H$ the graphic matroid of H.

REDUCTION TO MATROID PARITY

- $\mathcal{M}_H = (\mathrm{E_H}, \mathcal{F})$, where $\mathcal{F} = \{E' \subseteq \mathrm{E_H} \mid H(W, E') \text{ is acyclic}\}$

Matroid parity Instance

- Define a (multi-)set $\mathcal{P} \subseteq \binom{\mathrm{E_H}}{2}$, containing pairs of edges in H: for each vertex $u \in V \setminus W$, with neighbors x,y and z, add the pair {xy, yz} to $\mathcal{P}$.
- Let $|V|=n$ and t the parameter for the Special Disjoint FVS input. Set $k=n-|W|-t$.

- Show that G has a feedback vertex set $S \subseteq V \setminus W$ of size at most t $\Leftrightarrow$ *there is a subset* $\mathcal{Z} \subseteq \mathcal{P}$ of k disjoint pairs, such that $\cup_{X \in Z} X$ *is independent in* $\mathcal{M}_H$.

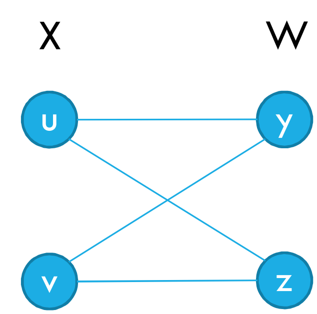
REDUCTION TO MATROID PARITY (CONT.)

- Define a bijection between subsets of V\W and subsets of $\mathcal{P}$.

For a set X $\subseteq$ V\W, define $\mathcal{Z}_X$={P_u | u $\in X$}
(This is a bijection since $\mathcal{P}$ is a multi-set.)

Claim: For any X $\subseteq$ V\W, the graph G[W∪ X] is acyclic
$\Leftrightarrow \cup_{P_{u \in Z_X}} P_u$ is independent in $\mathcal{M}_H$, and the pairs in $\mathcal{Z}_X$ are disjoint.

(i) $\Rightarrow$
(a) Suppose there are two pairs in $\mathcal{Z}_X$ that contain the same edge yz in H. Then u,y,v,z is a cycle in G[W ∪ X].



REDUCTION TO MATROID PARITY (CONT.)

Claim: For any X $\subseteq$ V\W, the graph G[W∪ X] is acyclic
$\Leftrightarrow \cup_{P_{u \in Z_X}} P_u$ is independent in $\mathcal{M}_H$, and the pairs in $\mathcal{Z}_X$ are disjoint.

(i) $\Rightarrow$

(b) Suppose the pairs in $\mathcal{Z}_X$ are disjoint, but $\cup_{P_{u \in Z_X}} P_u$ is not independent in $\mathcal{M}_H$. Then there is a cycle in H: C=$c_1 c_2 \cdots c_l$

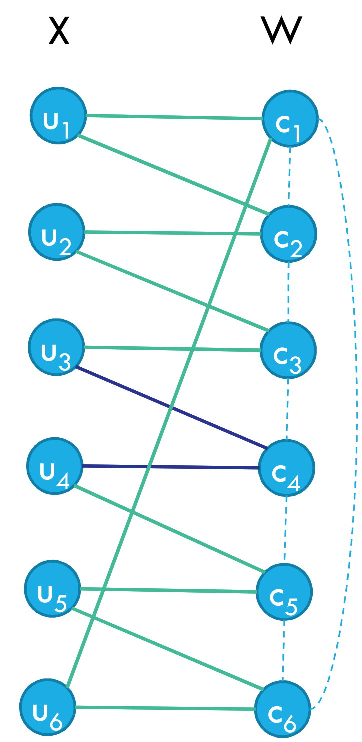
- For each 0 $\leq$ i $<$ l, let u_i be the vertex in X such that the pair corresponding to u_i contains the edge $c_i c_{i+1}$.
- Since the pairs in $\mathcal{Z}_X$ are disjoint, u_i is uniquely defined and adjacent to both c_i and c_{i+1}. The vertex u_l is adjacent to c_l and c_1.

REDUCTION TO MATROID PARITY (CONT.)

Claim: For any X ⊆ V\W, the graph G[W∪ X] is acyclic $\Leftrightarrow$ $\bigcup_{P_{u \in Z_X}} P_u$ is independent in $\mathcal{M}_H$, and the pairs in $\mathcal{Z}_X$ are disjoint.

(i) $\Rightarrow$

(b) The vertices $u_1, u_2, \cdots, u_l$ are not necessarily distinct, but cannot be all equal (C is a simple cycle on at least three edges, while any u_i corresponds to two edges). Hence, there is i such that $u_i \neq u_{i+1}$.



C=$c_1 c_2 \cdots c_6$ $\quad$ $\mathrm{i} = 3$

REDUCTION TO MATROID PARITY (CONT.)

Claim: For any X ⊆ V\W, the graph G[W∪ X] is acyclic ⟺ $\cup_{P_{u \in Z_X}} P_u$ is independent in $\mathcal{M}_H$, and the pairs in $\mathcal{Z}_X$ are disjoint.

(ii) $\Leftarrow$

Suppose that G[W∪ X] contains a cycle C. Let $E_C = \cup_{P_{u \in Z_C}} P_u$.

- If the pairs in $\mathcal{Z}_C$ are not disjoint we are done; else it can be shown that the subgraph induced by E_C contains a cycle. ■

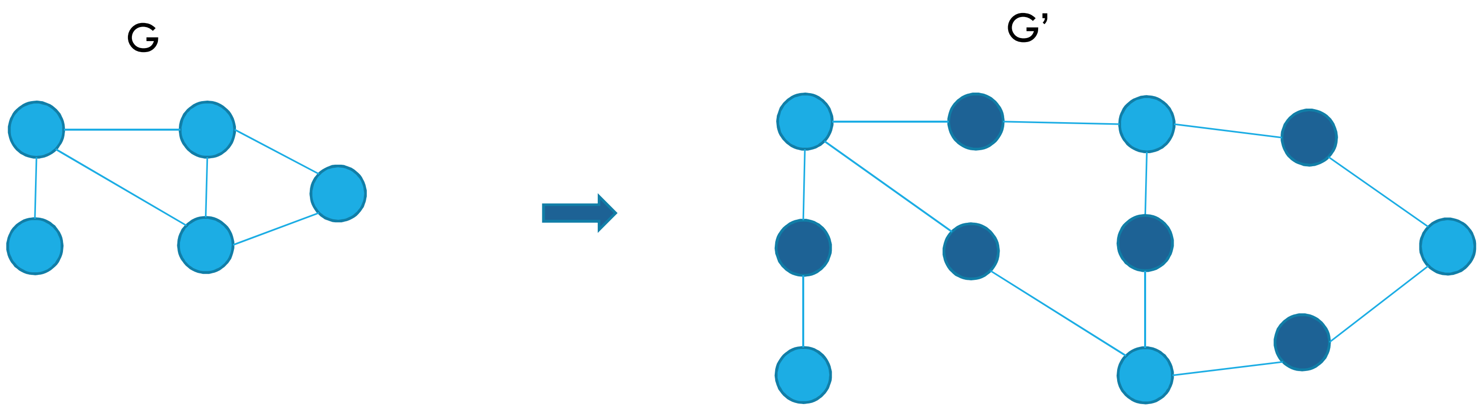
Observation: If there is an FVS S ⊆ V\W of size at most t, then X= V\(W∪S) is of size at least n-|W|-t=k such that G[W∪ X] is acyclic, and vice versa.

FVS IN SUBCUBIC GRAPHS (CONT.)

Use a polynomial time algorithm for Matroid Parity to solve Special Disjoint FVS.

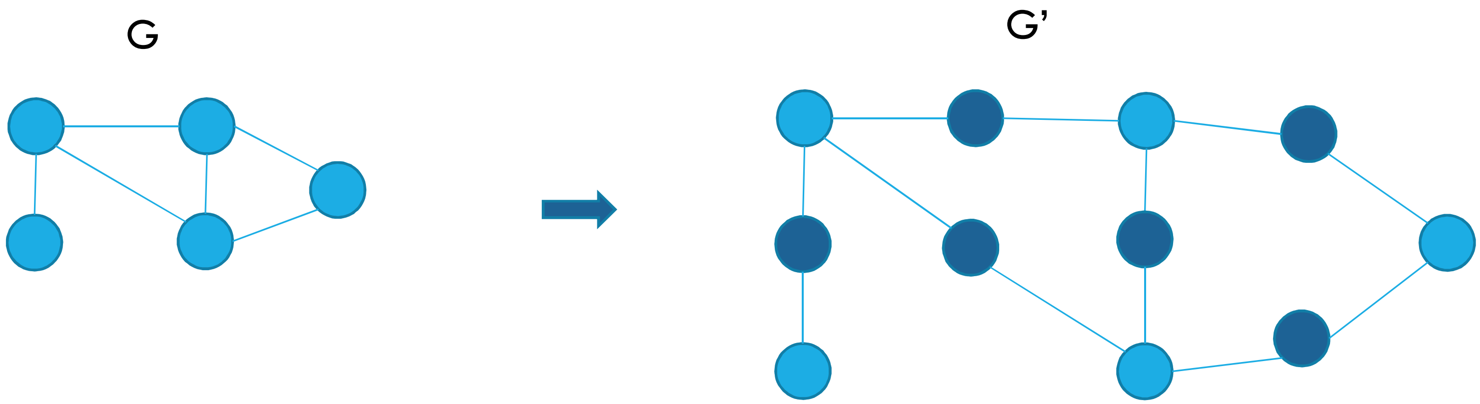
Theorem: FVS admits a polynomial-time algorithm on subcubic graphs.

Proof:



FVS IN SUBCUBIC GRAPHS (CONT.)

Theorem: FVS admits a polynomial-time algorithm on subcubic graphs.



- Let W be the set of new vertices.
- A subset $S \subseteq V(G)$ is FVS of G $\Leftrightarrow$ S is FVS of G’

Run the polynomial time algorithm on the Special Disjoint FVS instance (G’,W) ■

USE OF MATROIDS IN FPT ALGORITHMS

- Odd Cycle Transversal
- Long Directed Cycle
- Longest Path
- Almost 2-SAT
- Parameterized ℓ-Matroid Intersection
- Multiway Cut with a bounded number of terminals
- Many more…


Thank You