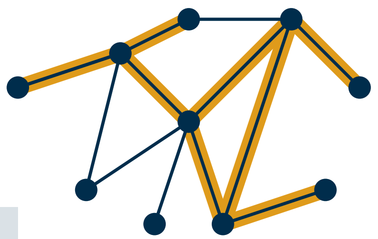
Parameterized Algorithms

Exercise 4 – Sheet 3/4, ILPs, and Coloring

Elly, Jean-Pierre, Wendy

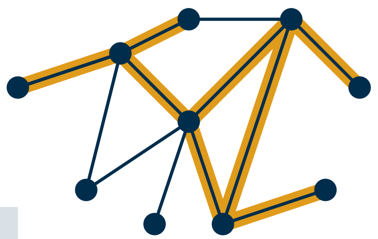
Sheet 3 – Odd Subgraph



Find a subgraph with exactly k edges where every vertex has odd degree.

$|\boldsymbol{E}| < \boldsymbol{k} \cdot (2\boldsymbol{k} - 1) \Rightarrow$ we already have a small kernel

Sheet 3 – Odd Subgraph

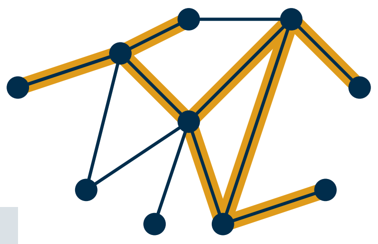


Find a subgraph with exactly k edges where every vertex has odd degree.

$|E| < k \cdot (2k - 1) \Rightarrow$ we already have a small kernel

$|E| \geq k \cdot (2k - 1)$

Sheet 3 – Odd Subgraph



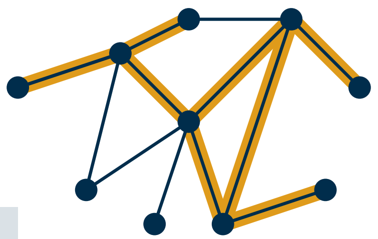
Find a subgraph with exactly k edges where every vertex has odd degree.

$\boldsymbol{|E| < k \cdot (2k-1)}$ $\Rightarrow$ we already have a small kernel

$\boldsymbol{|E| \geq k \cdot (2k-1)}$

- max deg $\leq k \Rightarrow$ find matching
  - greedily pick an edge, this removes at most $2 \cdot (k-1)$ other edges

Sheet 3 – Odd Subgraph



Find a subgraph with exactly k edges where every vertex has odd degree.

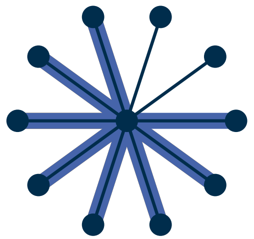
$|E| < k \cdot (2k-1) \Rightarrow$ we already have a small kernel

$|E| \geq k \cdot (2k-1)$

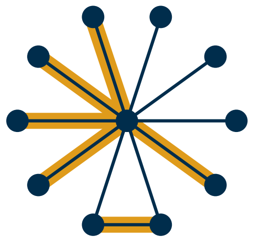
- max deg $\leq k \Rightarrow$ find matching
  - greedily pick an edge, this removes at most $2 \cdot (k-1)$ other edges
- max deg $> k$

k odd: pick star

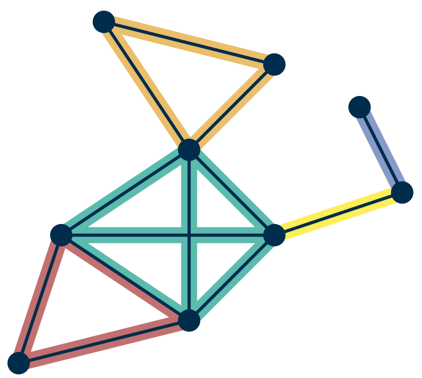
k even and G is star: no solution



k even and G is **no** star: pick non incident edge

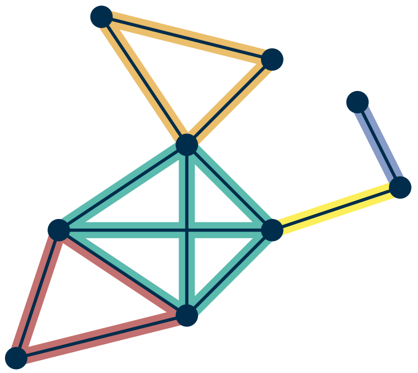


Sheet 3 – Edge Clique Cover



Find a set of at most k cliques that cover all edges.

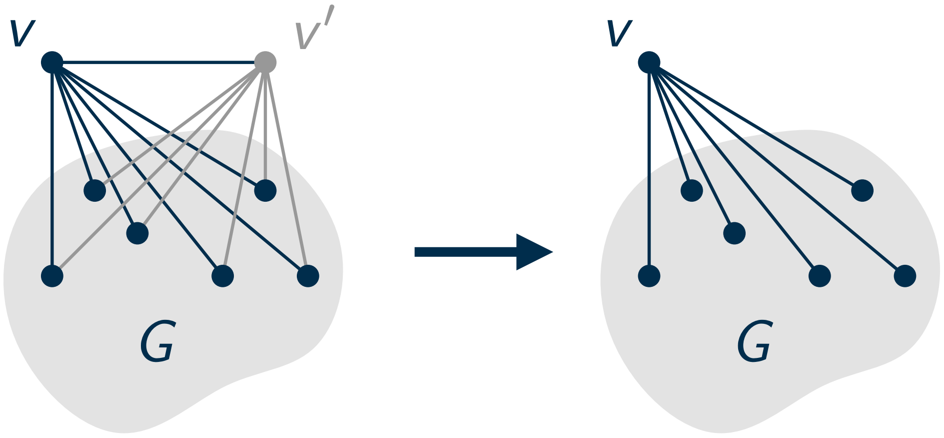
Sheet 3 – Edge Clique Cover



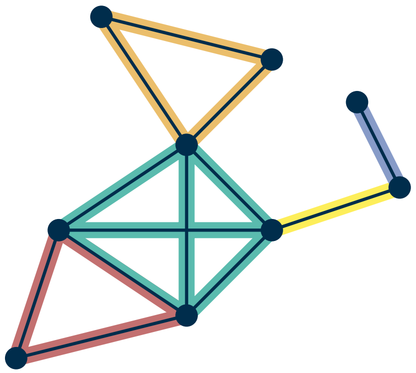
Find a set of at most k cliques that cover all edges.

Reduction Rule

- merge **true twins**



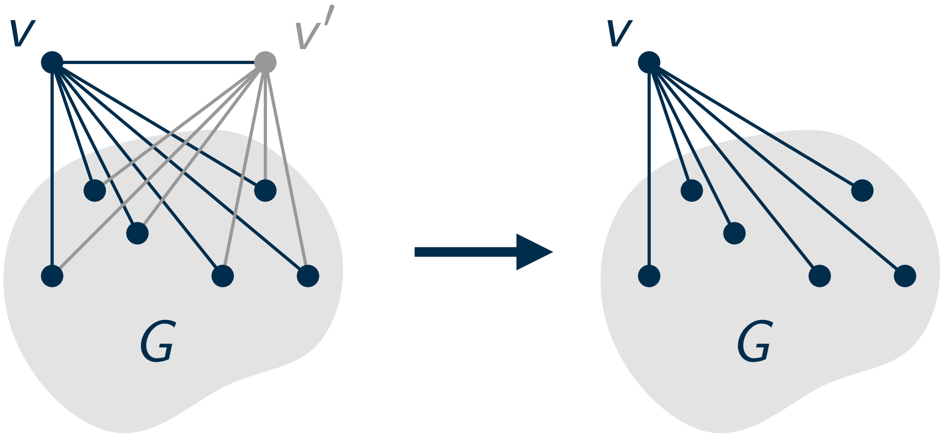
Sheet 3 – Edge Clique Cover



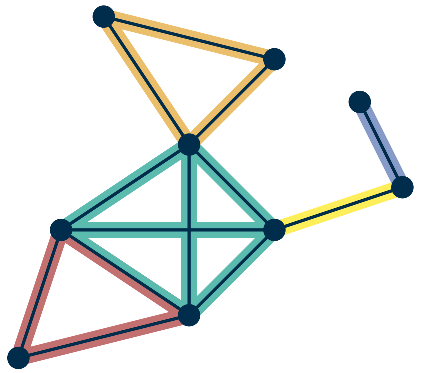
Find a set of at most k cliques that cover all edges.

Reduction Rule

- merge **true twins**
  - G covered by k Cliques $\Leftrightarrow$
    $G - v'$ coverd by k Cliques



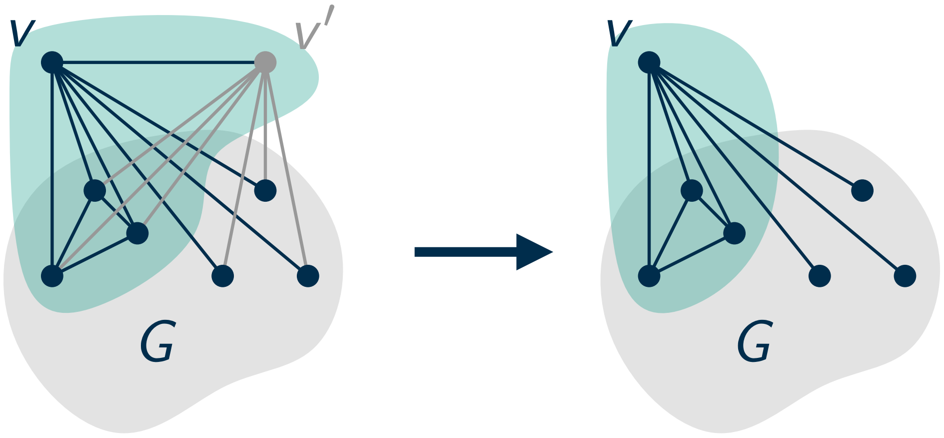
Sheet 3 – Edge Clique Cover



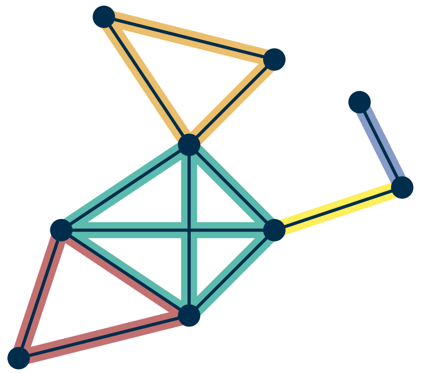
Find a set of at most k cliques that cover all edges.

Reduction Rule

- merge **true twins**
  - G covered by k Cliques $\Leftrightarrow$ $G - v'$ coverd by k Cliques



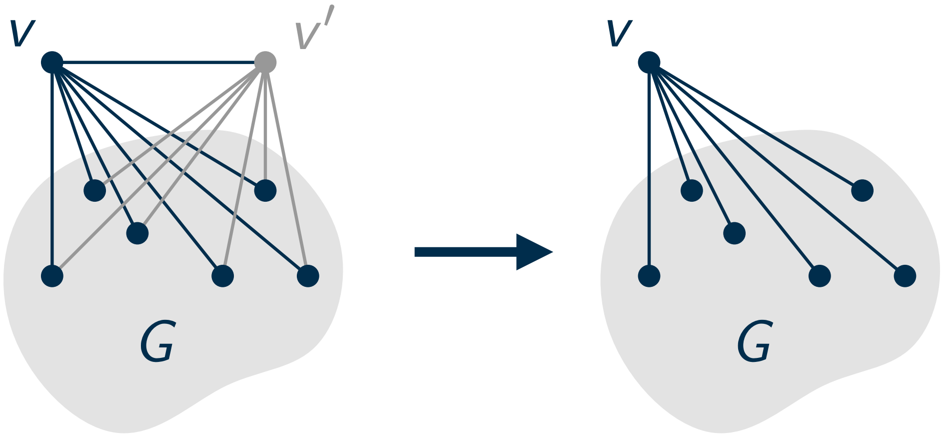
Sheet 3 – Edge Clique Cover



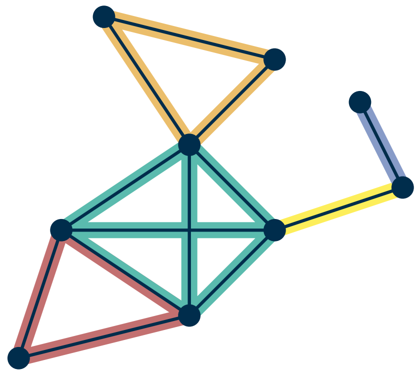
Find a set of at most k cliques that cover all edges.

Reduction Rule

- merge **true twins**
  - G covered by k Cliques $\Leftrightarrow$ $G - v'$ coverd by k Cliques
  - reduce k if vv' is isolated edge



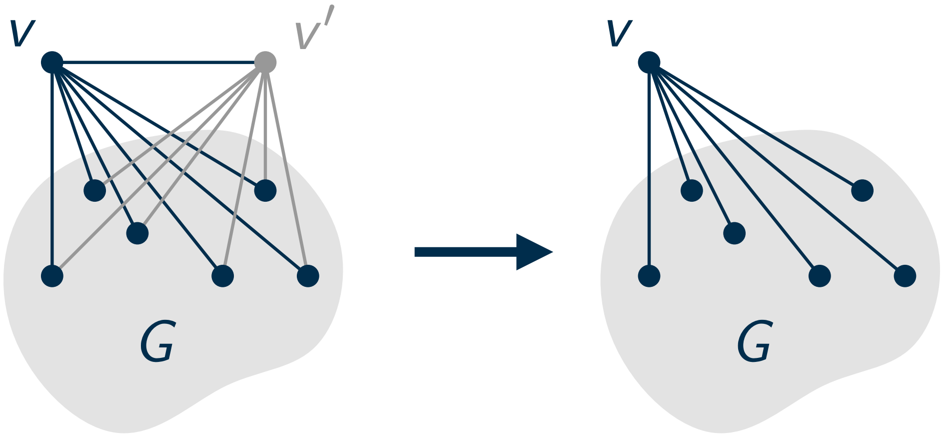
Sheet 3 – Edge Clique Cover



Find a set of at most k cliques that cover all edges.

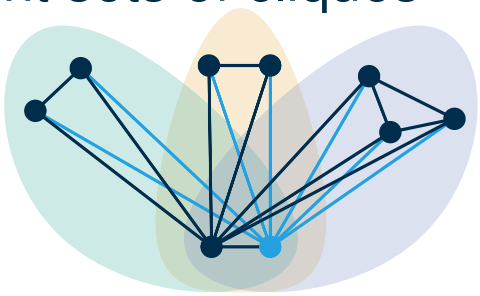
Reduction Rule

- merge **true twins**
  - G covered by k Cliques $\Leftrightarrow$ $G - v'$ coverd by k Cliques
  - reduce k if vv' is isolated edge

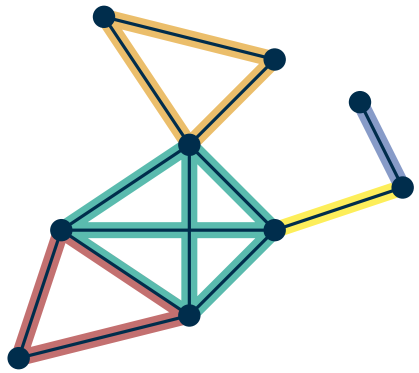


Small Kernel after Reduction

- different vertices have to appear in different sets of cliques
  - otherwise they are true twins



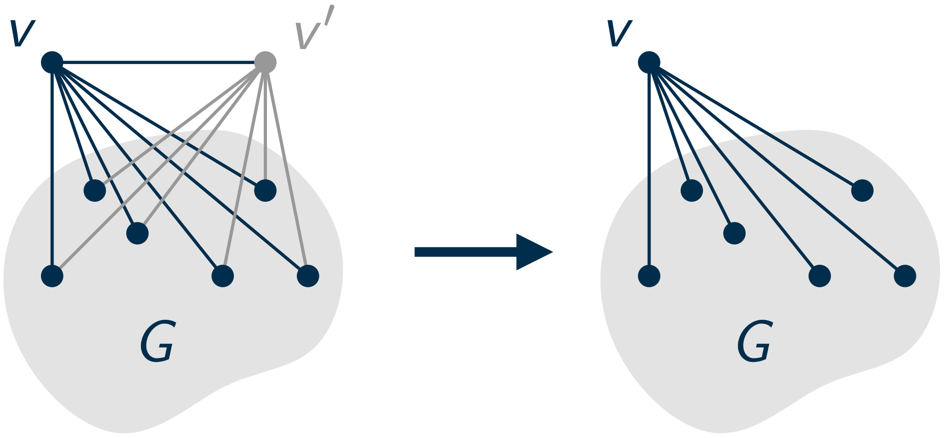
Sheet 3 – Edge Clique Cover



Find a set of at most k cliques that cover all edges.

Reduction Rule

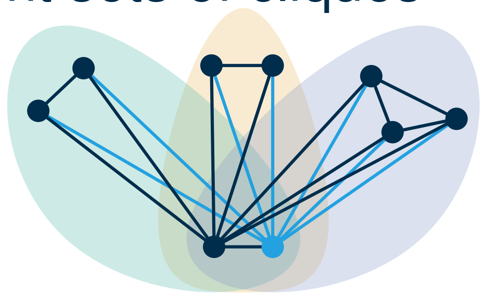
- merge **true twins**
  - G covered by k Cliques $\Leftrightarrow$ $G - v'$ coverd by k Cliques
  - reduce k if vv' is isolated edge



Small Kernel after Reduction

- different vertices have to appear in different sets of cliques
  - otherwise they are true twins

$\Rightarrow$ a solution with at most k cliques has at most 2^k vertices

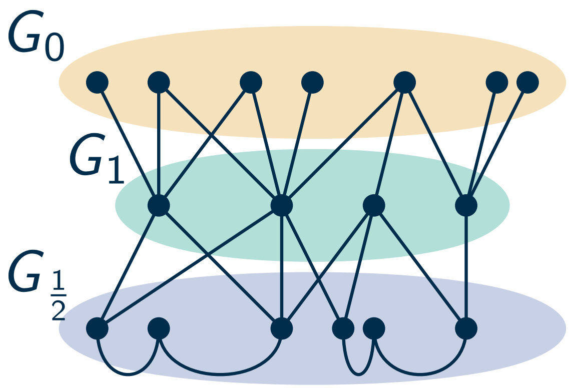


Sheet 3 – VC on Bipartite Graphs

The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

Sheet 3 – VC on Bipartite Graphs

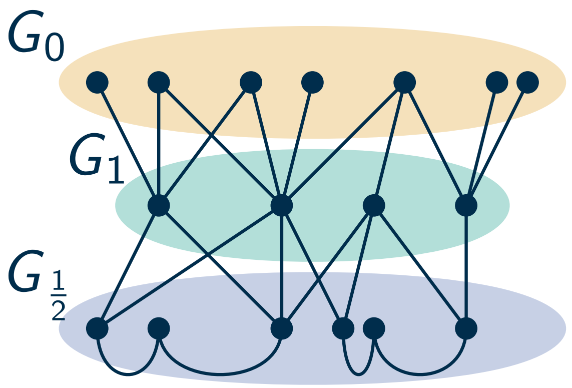
The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).



Sheet 3 – VC on Bipartite Graphs

The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

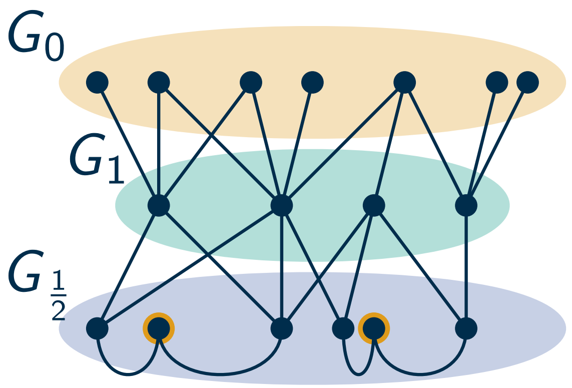
- Set the smaller bipartite component in $G_{\frac{1}{2}}$ to 1



Sheet 3 – VC on Bipartite Graphs

The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

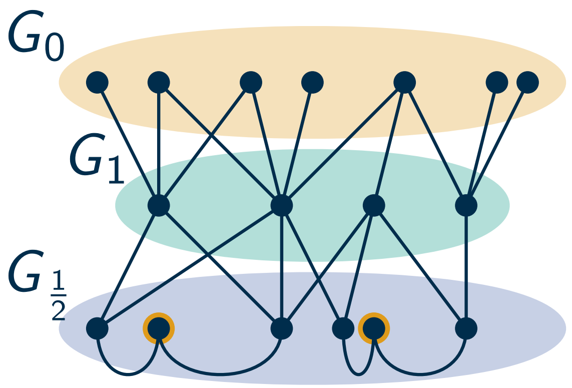
- Set the smaller bipartite component in $G_{\frac{1}{2}}$ to 1



Sheet 3 – VC on Bipartite Graphs

The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

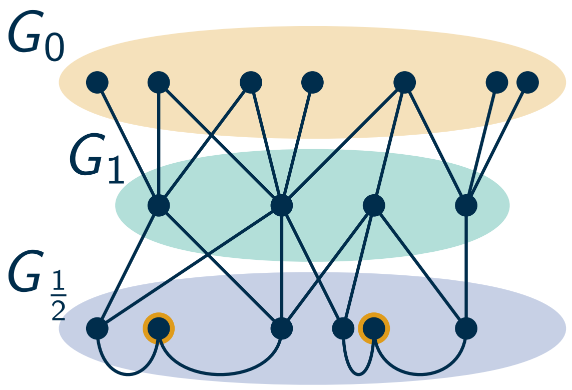
- Set the smaller bipartite component in $G_{\frac{1}{2}}$ to 1
  - edges in old assignment remain covered



Sheet 3 – VC on Bipartite Graphs

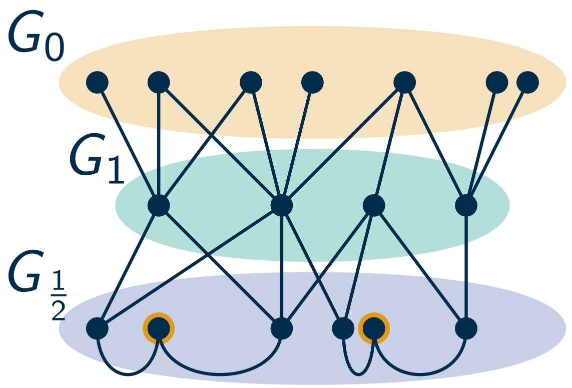
The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

- Set the smaller bipartite component in $G_{\frac{1}{2}}$ to 1
  - edges in old assignment remain covered



In a bipartite graph: |min VC| = |max matching|

Sheet 3 – VC on Bipartite Graphs



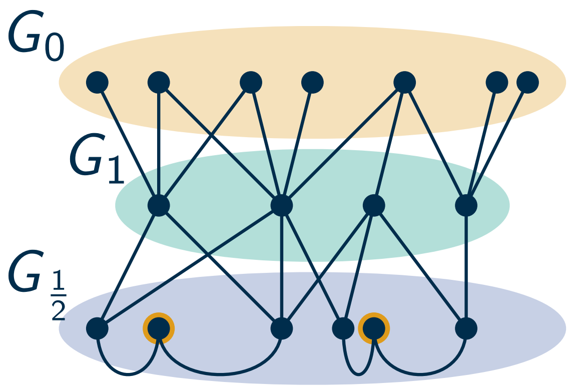
The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

- Set the smaller bipartite component in $G_{\frac{1}{2}}$ to 1
  - edges in old assignment remain covered

In a bipartite graph: |min VC| = |max matching|

- LP relaxation of VC is dual to LP relaxation of matching

Sheet 3 – VC on Bipartite Graphs



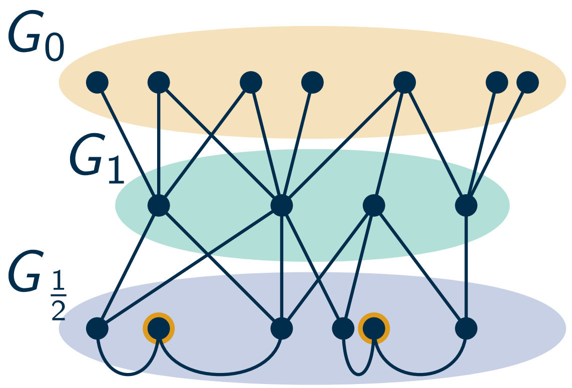
The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

- Set the smaller bipartite component in $G_{\frac{1}{2}}$ to 1
  - edges in old assignment remain covered

In a bipartite graph: |min VC| = |max matching|

- LP relaxation of VC is dual to LP relaxation of matching
- ILP VC $=$ LP VC $=$ LP matching $=$ ILP matching

Sheet 3 – VC on Bipartite Graphs



The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

- Set the smaller bipartite component in $G_{\frac{1}{2}}$ to 1
  - edges in old assignment remain covered

In a bipartite graph: |min VC| = |max matching|

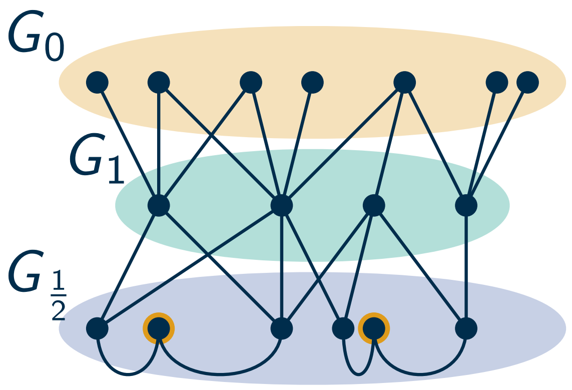
- LP relaxation of VC is dual to LP relaxation of matching
- ILP VC $=$ LP VC $=$ LP matching $=$ ILP matching

Incidence matrix $A \in \mathbb{R}^{n \times m}$

Sheet 3 – VC on Bipartite Graphs

The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

- Set the smaller bipartite component in $G_{\frac{1}{2}}$ to 1
  - edges in old assignment remain covered



Incidence matrix $A \in \mathbb{R}^{n \times m}$

$x_j = 1$: edge j is in matching

$y_i = 1$: vertex i is in VC

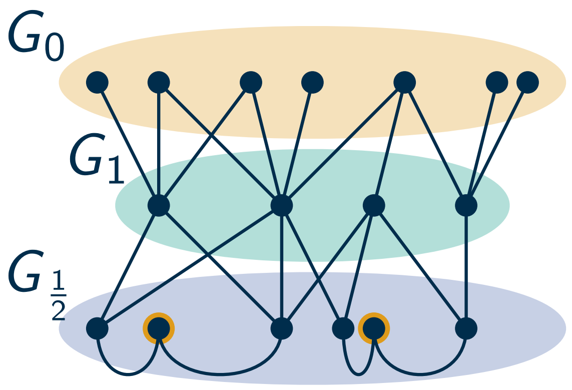
In a bipartite graph: |min VC| = |max matching|

- LP relaxation of VC is dual to LP relaxation of matching
- ILP VC $=$ LP VC $=$ LP matching $=$ ILP matching

Sheet 3 – VC on Bipartite Graphs

The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).

- Set the smaller bipartite component in $G_{\frac{1}{2}}$ to 1
  - edges in old assignment remain covered



In a bipartite graph: |min VC| = |max matching|

- LP relaxation of VC is dual to LP relaxation of matching
- ILP VC $=$ LP VC $=$ LP matching $=$ ILP matching

Incidence matrix $A \in \mathbb{R}^{n \times m}$
$x_j = 1$: edge j is in matching
$y_i = 1$: vertex i is in VC

$$\max \sum_{j=1}^{n} x_j \quad \text{with } Ax \leq \mathbf{1}$$

$$\min \sum_{i=1}^{n} y_i \quad \text{with } A^T y \geq \mathbf{1}$$

Sheet 3 – VC on Bipartite Graphs

VC can be solved in $O(m\sqrt{n})$ on bipartite graphs

- The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).
- In a bipartite graph: |min VC| = |max matching|

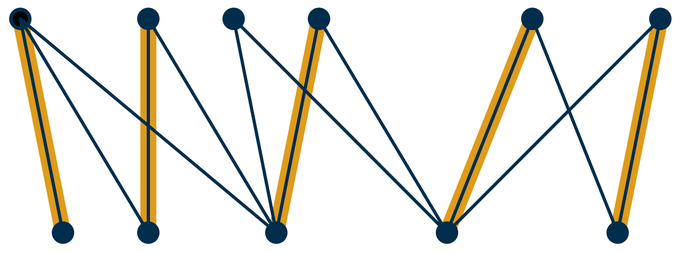
Sheet 3 – VC on Bipartite Graphs

VC can be solved in $O(m\sqrt{n})$ on bipartite graphs

- The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).
- In a bipartite graph: |min VC| = |max matching|

Reduction to 2-SAT

- first, compute matching in $O(m\sqrt{n})$



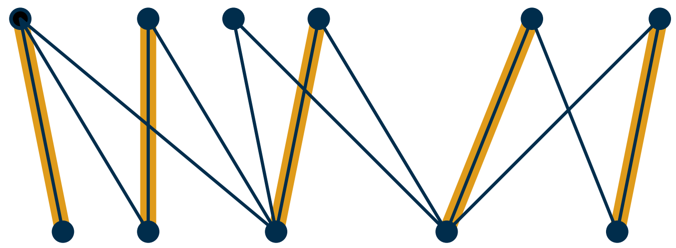
Sheet 3 – VC on Bipartite Graphs

VC can be solved in $O(m\sqrt{n})$ on bipartite graphs

- The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).
- In a bipartite graph: |min VC| = |max matching|

Reduction to 2-SAT

- first, compute matching in $O(m\sqrt{n})$
- Idea: for every matching edge exactly one vertex has to be choosen



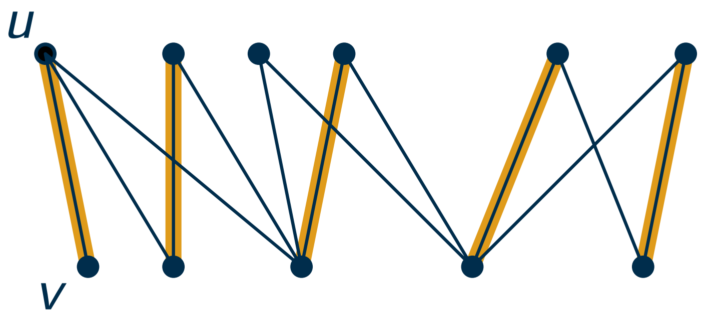
Sheet 3 – VC on Bipartite Graphs

VC can be solved in $O(m\sqrt{n})$ on bipartite graphs

- The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).
- In a bipartite graph: |min VC| = |max matching|

Reduction to 2-SAT

- first, compute matching in $O(m\sqrt{n})$
- Idea: for every matching edge exactly one vertex has to be choosen
  - one vertex per matching edge uv: $(u \vee v), (\neg u \vee \neg v)$



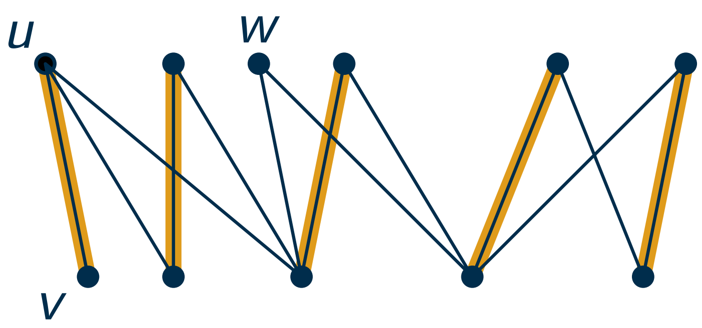
Sheet 3 – VC on Bipartite Graphs

VC can be solved in $O(m\sqrt{n})$ on bipartite graphs

- The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).
- In a bipartite graph: |min VC| = |max matching|

Reduction to 2-SAT

- first, compute matching in $O(m\sqrt{n})$
- Idea: for every matching edge exactly one vertex has to be choosen
  - one vertex per matching edge uv: $(u \vee v), (\neg u \vee \neg v)$
  - no unmatched vertex w is picked: $(\neg w)$



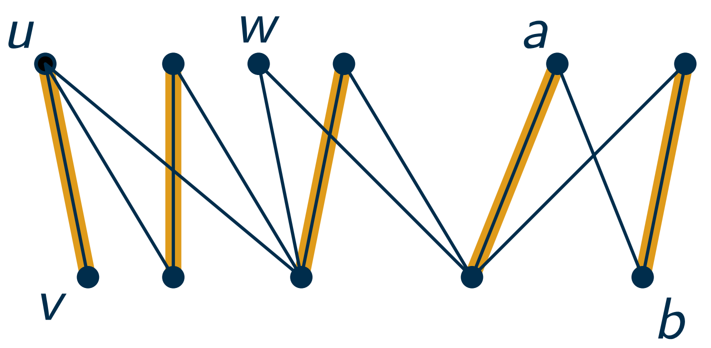
Sheet 3 – VC on Bipartite Graphs

VC can be solved in $O(m\sqrt{n})$ on bipartite graphs

- The LP relaxation of VC on bipartite graphs has an optimal integer solution (using only 0 or 1).
- In a bipartite graph: |min VC| = |max matching|

Reduction to 2-SAT

- first, compute matching in $O(m\sqrt{n})$
- Idea: for every matching edge exactly one vertex has to be choosen
  - one vertex per matching edge uv: $(u \vee v), (\neg u \vee \neg v)$
  - no unmatched vertex w is picked: $(\neg w)$
  - all edges ab are covered: $(a \vee b)$



Sheet 3 – Closest String

CLOSEST STRING
Given: A list L of ℓ strings. Find the minimum k and string s, such that s has Hamming distance at most k to each string.

```
dbaccbbbcbacccbbadbb
abbccabbcbacccabadbb
bbaccabccbacccabadbb
```

Sheet 3 – Closest String

Closest String
Given: A list L of ℓ strings. Find the minimum k and string s, such that s has Hamming distance at most k to each string.

```
dbaccbbbcbacccbbadbb
abbccabbcbacccabadbb
bbaccabccbacccabadbb
--------------------
dbaccabbcbacccabadbb
```

$k = 2$

Sheet 3 – Closest String

CLOSEST STRING
Given: A list L of ℓ strings. Find the minimum k and string s, such that s has Hamming distance at most k to each string.

dbaccbbbcbacccbbadbb
abbccabbcbacccabadbb
bbaccabccbacccabadbb

dbaccabbcbacccabadbb
$k = 2$

- pick initial candidate s^*
- if *every* string in L hast distance $\leq k$: YES
- if *some* string in L hast distance $> 2k$: NO

Sheet 3 – Closest String

CLOSEST STRING
Given: A list L of ℓ strings. Find the minimum k and string s, such that s has Hamming distance at most k to each string.

dbaccbbbcbacccbbadbb
abbccabbcbacccabadbb
bbaccabccbacccabadbb

dbaccabbcbacccabadbb
$k = 2$

- pick initial candidate s^*
- if *every* string in L hast distance $\leq k$: YES
- if *some* string in L hast distance $> 2k$: NO
- find string with distance $k < \delta \leq 2k$
  - branch about δ decisions

Sheet 3 – Closest String

Closest String
Given: A list L of ℓ strings. Find the minimum k and string s, such that s has Hamming distance at most k to each string.

```
dbaccbbbcbacccbbadbb
abbccabbcbacccabadbb
bbaccabccbacccabadbb
--------------------
dbaccabbcbacccabadbb
```

$k = 2$

- pick initial candidate s^*
- if *every* string in L hast distance $\leq k$: YES
- if *some* string in L hast distance $> 2k$: NO
- find string with distance $k < \delta \leq 2k$
  - branch about δ decisions

How to choose initial candidate?

Sheet 3 – Closest String

Closest String
Given: A list L of ℓ strings. Find the minimum k and string s, such that s has Hamming distance at most k to each string.

```
dbaccbbbcbacccbbadbb
abbccabbcbacccabadbb
bbaccabccbacccabadbb
--------------------
dbaccabbcbacccabadbb
```

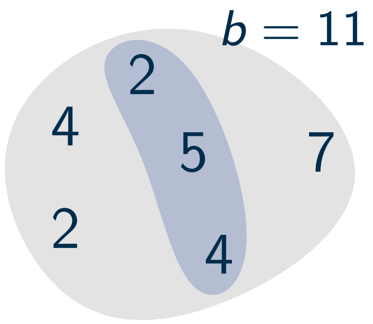
$k = 2$

- pick initial candidate s^*
- if *every* string in L hast distance $\leq k$: YES
- if *some* string in L hast distance $> 2k$: NO
- find string with distance $k < \delta \leq 2k$
  - branch about δ decisions

How to choose initial candidate?

Which string do we branch over?

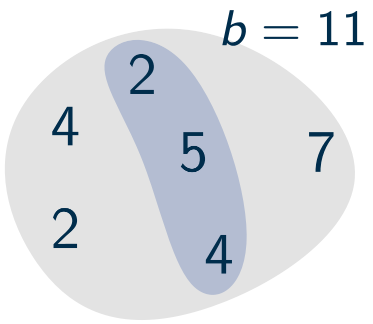
Sheet 4



Formulate ILP

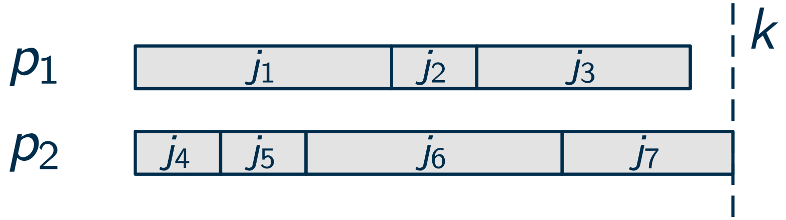
- SUBSET SUM: find sub-multiset of size b

Sheet 4



Formulate ILP

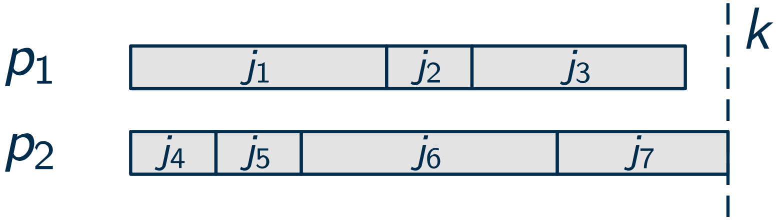
- SUBSET SUM: find sub-multiset of size b
- SCHEDULING: find job assignment with makespan k

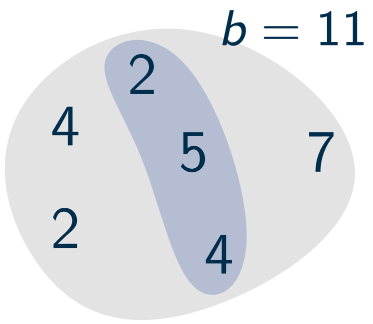


Sheet 4

Formulate ILP

- SUBSET SUM: find sub-multiset of size b
- SCHEDULING: find job assignment with makespan k



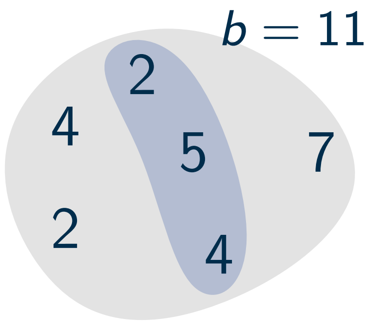


MAX SAT

given CNF, satisfy at least p clauses

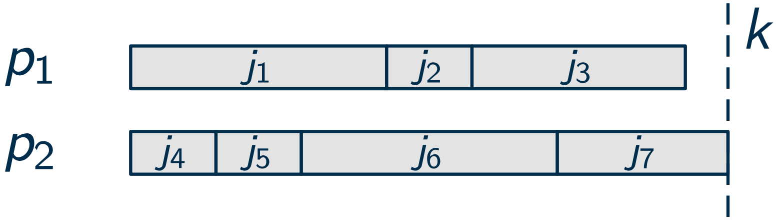
$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_4 \vee \neg x_5)$$

Sheet 4



Formulate ILP

- SUBSET SUM: find sub-multiset of size b
- SCHEDULING: find job assignment with makespan k



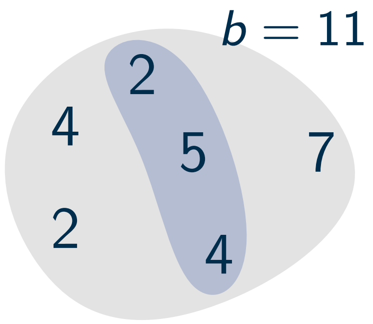
MAX SAT

given CNF, satisfy at least p clauses

- find kernel of size $O(p)$

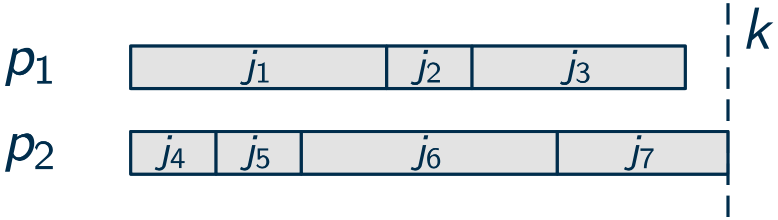
$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_4 \vee \neg x_5)$$

Sheet 4



Formulate ILP

- SUBSET SUM: find sub-multiset of size b
- SCHEDULING: find job assignment with makespan k



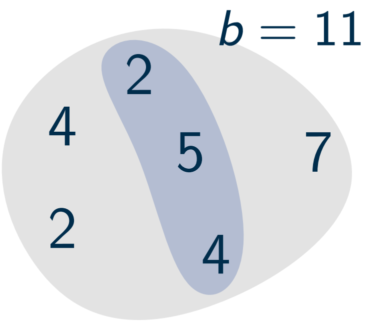
MAX SAT

given CNF, satisfy at least p clauses

- find kernel of size $O(p)$
- FPT algo for parameter $k = p - \frac{m}{2}$

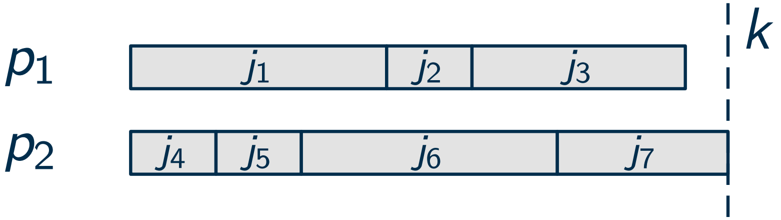
$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_4 \vee \neg x_5)$$

Sheet 4



Formulate ILP

- SUBSET SUM: find sub-multiset of size b
- SCHEDULING: find job assignment with makespan k



MAX SAT

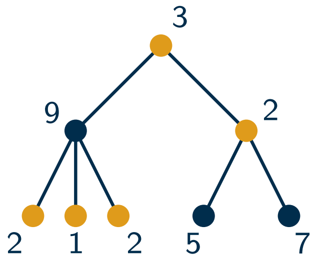
given CNF, satisfy at least p clauses

- find kernel of size $O(p)$
- FPT algo for parameter $k = p - \frac{m}{2}$

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_4 \vee \neg x_5)$$

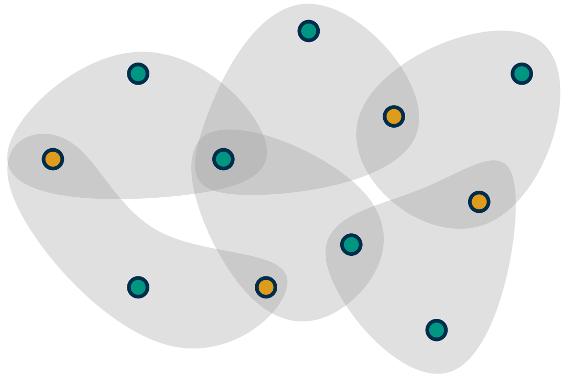
WEIGHTED VERTEX COVER on Trees

- write down DP for weighted VC on trees



Probabilistic Method

HYPERGRAPH 2-COLORING
Given hypergraph where each hyperedge has size exactly k. Is there an assignment of 2 colors such that each edge is colorful?



- **Idea:** assign every vertex a random color
- consider the probability that and edge e is single-colored

$$\mathbb{P}[\exists e \text{ that is single-colored}] \leq \sum_{e \in E} \mathbb{P}[e \text{ is single-colored}] = |E| \cdot 2^{1-k}$$

- $|E| < 2^{k-1} \Rightarrow \mathbb{P}[\exists e \text{ that is single-colored}] < 1 \Rightarrow$ there exists a colorful assignment

$\mathbb{P}[X] > 0 \Rightarrow$ there exists an event such that X holds

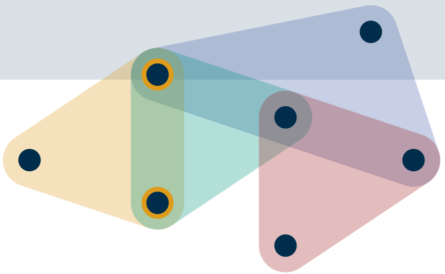
ILPs and Coloring

Formulate the following problems as an ILP:

ILPs and Coloring

Formulate the following problems as an ILP:

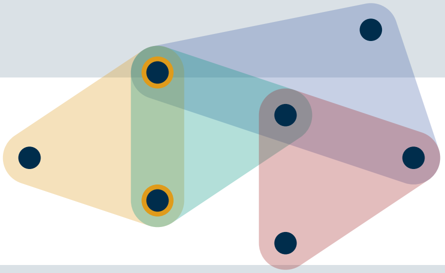
HITTING SET
Given: Collection of sets S and parameter k. Is there a subset of size at most k that hits every set in S?



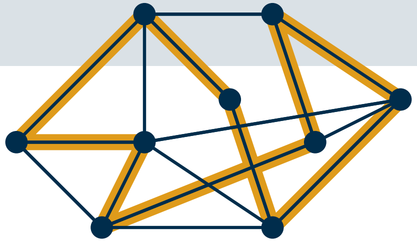
ILPs and Coloring

Formulate the following problems as an ILP:

HITTING SET
Given: Collection of sets S and parameter k. Is there a subset of size at most k that hits every set in S?



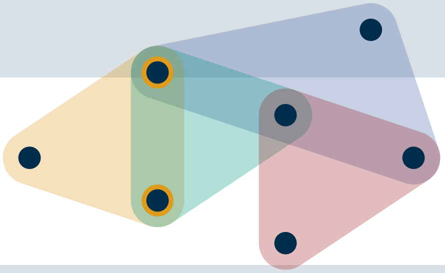
HAMILTON CYCLE
Given: Graph G. Is there a cycle in G that containes every vertex?



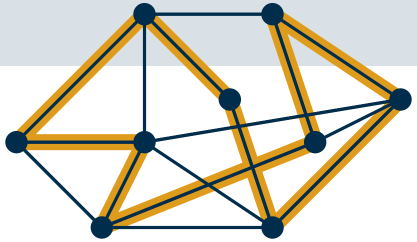
ILPs and Coloring

Formulate the following problems as an ILP:

HITTING SET
Given: Collection of sets S and parameter k. Is there a subset of size at most k that hits every set in S?



HAMILTON CYCLE
Given: Graph G. Is there a cycle in G that containes every vertex?

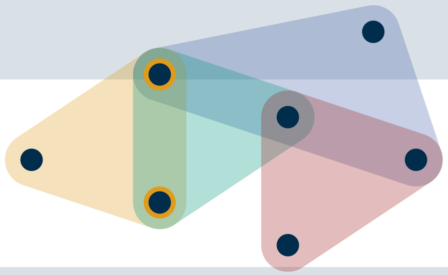


Show that the following problems are in FPT using color coding:

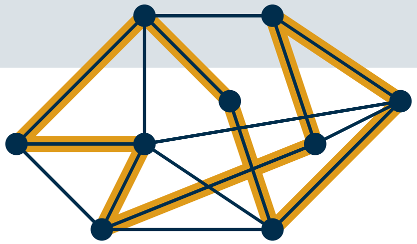
ILPs and Coloring

Formulate the following problems as an ILP:

HITTING SET
Given: Collection of sets S and parameter k. Is there a subset of size at most k that hits every set in S?

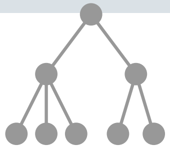
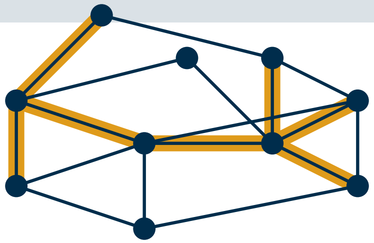


HAMILTON CYCLE
Given: Graph G. Is there a cycle in G that containes every vertex?



Show that the following problems are in FPT using color coding:

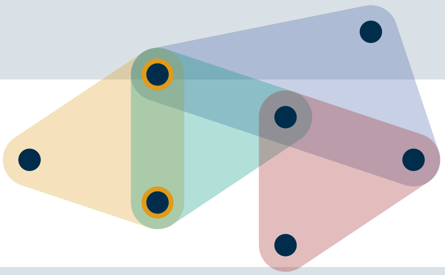
TREE SUBGRAPH ISOMORPHISM
Given: Graph G, parameter k, tree T with $|T| = k$. Does G contain T as a subgraph?

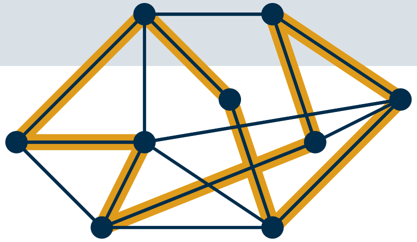
ILPs and Coloring

Formulate the following problems as an ILP:

HITTING SET
Given: Collection of sets S and parameter k. Is there a subset of size at most k that hits every set in S?

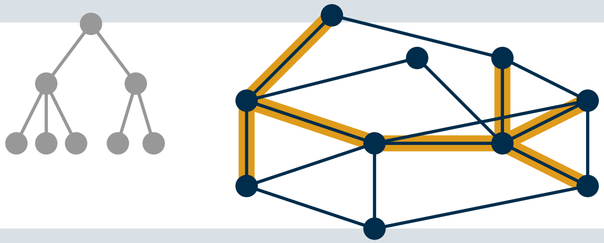


HAMILTON CYCLE
Given: Graph G. Is there a cycle in G that containes every vertex?

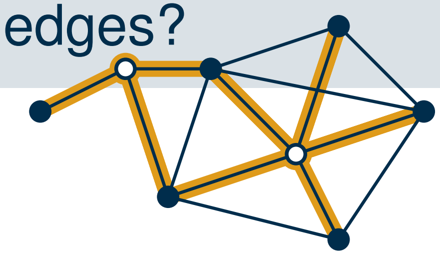


Show that the following problems are in FPT using color coding:

TREE SUBGRAPH ISOMORPHISM
Given: Graph G, parameter k, tree T with $|T| = k$. Does G contain T as a subgraph?

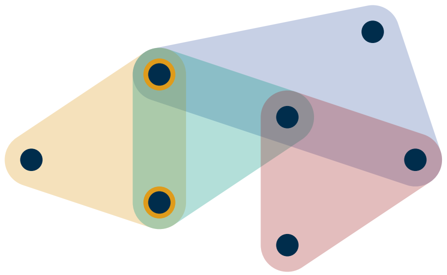


PARTIAL VERTEX COVER
Given: Graph G, parameter k and upper bound s. Is there a subset of at most s vertices that hits at least k edges?



Solution ILP

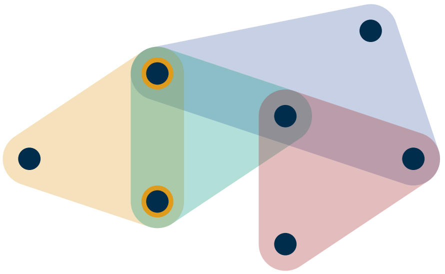
Hitting Set
Given: Collection of sets S and parameter k. Is there a subset of size at most k that hits every set in S?



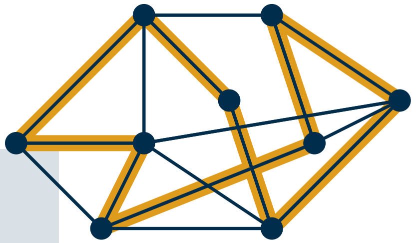
- one variable for every element $v = 1 \Leftrightarrow v$ is in hitting set
- **every set is hit:** for all $s \in S : \sum_{w \in s} w \geq 1$.
- minimize the sum of all variables

Solution ILP

Hitting Set
Given: Collection of sets S and parameter k. Is there a subset of size at most k that hits every set in S?



- one variable for every element $v = 1 \Leftrightarrow v$ is in hitting set
- **every set is hit:** for all $s \in S : \sum_{w \in s} w \geq 1$.
- minimize the sum of all variables

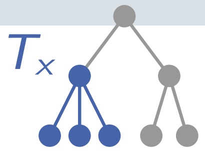


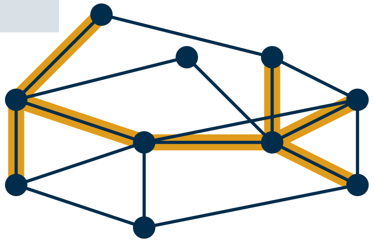
Hamilton Cycle
Given: Graph G. Is there a cycle in G that containes every vertex?

- encode permutation, $x_{v,i} = 1 \Leftrightarrow v$ is the ith vertex on the cycle
- **valid permutation:** for all vertices $w : \sum_j x_{w,j} = 1$ (same for timesteps)
- **valid cycle:** $x_{v,i} + x_{w,i+1} - e_{vw} \leq 1$ (e_{vw} is 1 when vw is an edge)

Solution Coloring

TREE SUBGRAPH ISOMORPHISM
Given: Graph G, parameter k, tree T with $|T| = k$. Does G contain T as a subgraph?

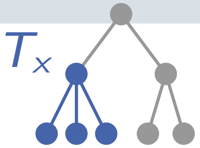


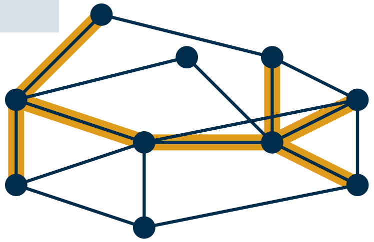


- **Coloful version:** does G contain T as a coloful subgraph?
- **DP:** for every vertex v: compute for which color sets T_x is a subgraph with v as root

Solution Coloring

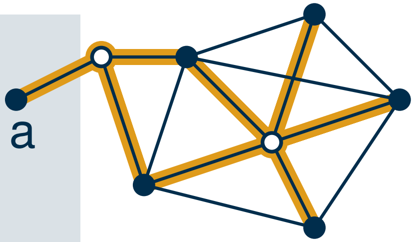
Tree Subgraph Isomorphism
Given: Graph G, parameter k, tree T with $|T| = k$. Does G contain T as a subgraph?





- **Coloful version:** does G contain T as a coloful subgraph?
- **DP:** for every vertex v: compute for which color sets T_x is a subgraph with v as root

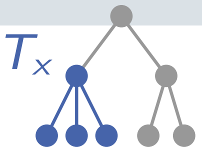
Partial Vertex Cover
Given: Graph G, parameter k and upper bound s. Is there a subset of at most s vertices that hits at least k edges?

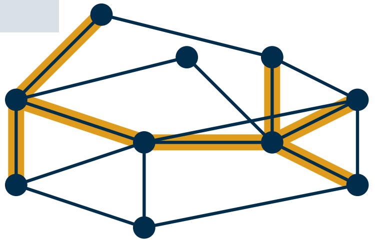


- **Color edges:** Can we select a set of s vertices that hits each color calss?

Solution Coloring

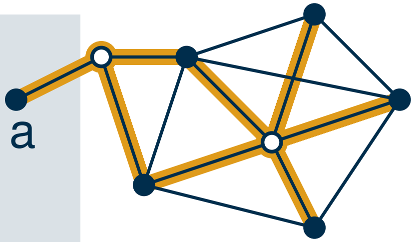
TREE SUBGRAPH ISOMORPHISM
Given: Graph G, parameter k, tree T with $|T| = k$. Does G contain T as a subgraph?

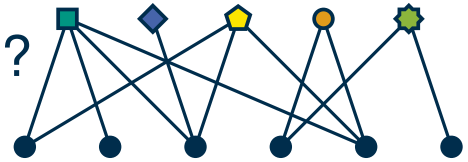




- **Coloful version:** does G contain T as a coloful subgraph?
- **DP:** for every vertex v: compute for which color sets T_x is a subgraph with v as root

PARTIAL VERTEX COVER
Given: Graph G, parameter k and upper bound s. Is there a subset of at most s vertices that hits at least k edges?





- **Color edges:** Can we select a set of s vertices that hits each color calss?
- Interpret a hyper/bipartite graph with at most 2^k vertices
- **DP:** For each combination of colors. Minimum number of vertices from $\{v_1, \ldots, v_i\}$ to cover this combination