Advanced topics on Algorithms

Luciano Gualà
www.mat.uniroma2.it/~guala/

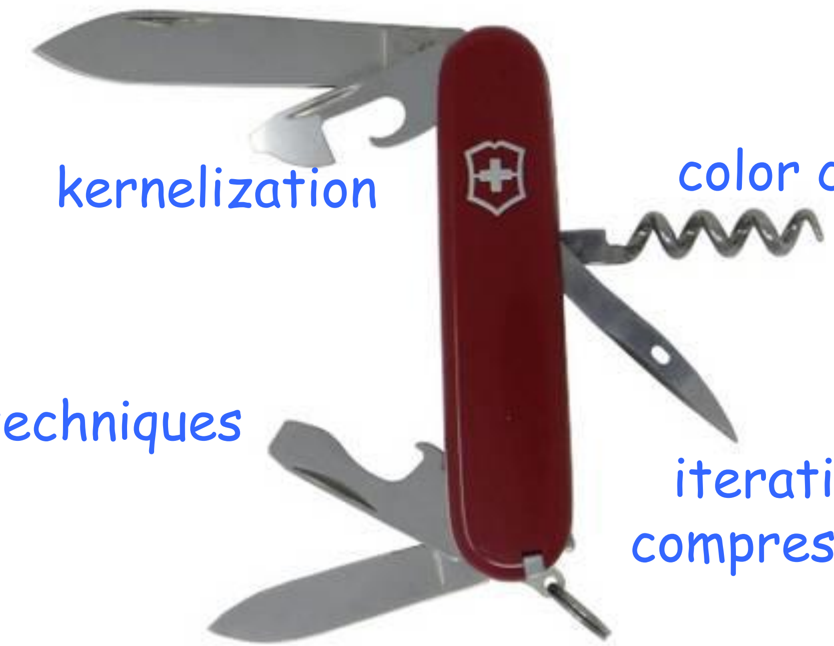
Parameterized algorithms
Episode III

Toolbox (to show a problem is FPT)

bounded-search trees

kernelization

color coding



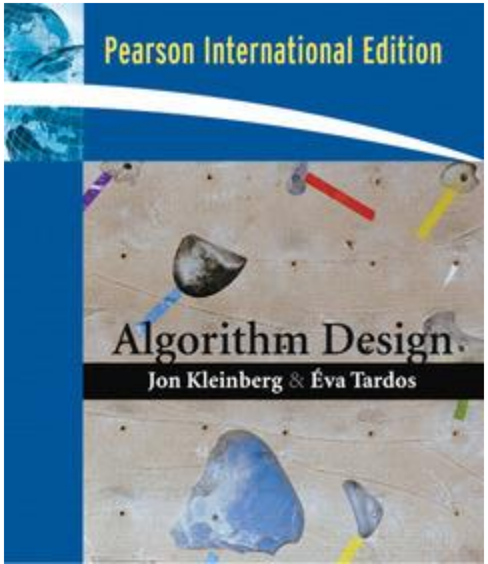
algebraic techniques

iterative compression

treewidth

Treewidth

reference
(Chapter 10.4)



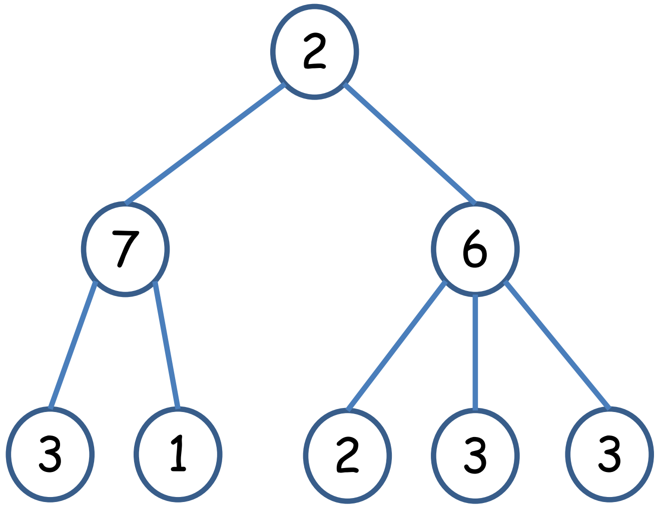
The party problem

problem: invite people to a party

maximize: total fun factor of the invited people

constraint: everyone should be having fun

⇒ do not invite a colleague and his direct boss at the same time!



input: a tree with weights on the nodes

goal: an independent set of maximum total weight

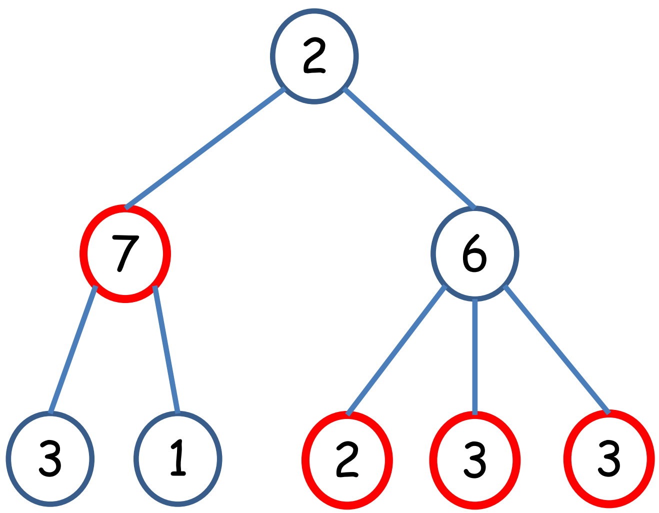
The party problem

problem: invite people to a party

maximize: total fun factor of the invited people

constraint: everyone should be having fun

⇒ do not invite a colleague and his direct boss at the same time!



OPT= 15

input: a tree with weights on the nodes

goal: an independent set of maximum total weight

weighted independent set on trees: a dynamic programming algorithm

Subproblems:

For each v of T:

- T_v: subtree of T rooted at v
- A[v]: weight of a maximum weighted IS of T_v
- B[v]: weight of a maximum weighted IS of T_v
  that does not contain v

goal: determine A[r] for the root r

v leaf: $A[v]=w_v$ $B[v]=0$

v internal node with children $u_1,\ldots,u_d$:

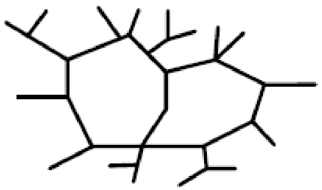
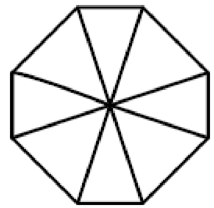
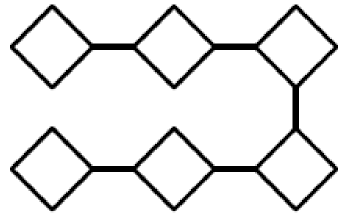
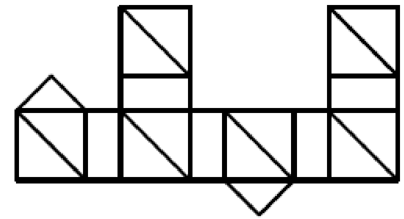
$$B[v]=\sum_{i=1}^{d} A[u_i]$$

$$A[v]=\max\{\ B[v],\ \ w_v+\sum_{i=1}^{d} B[u_i]\ \}$$

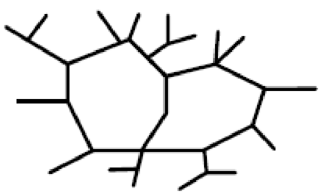
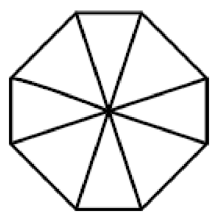
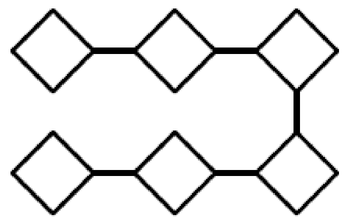
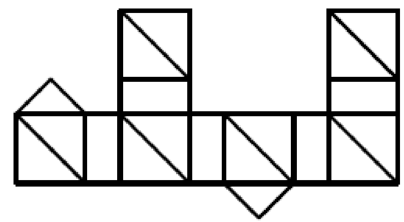
order for the subproblems: bottom up

Generalizing trees: How could we define that a graph is "treelike"?

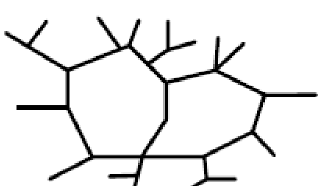
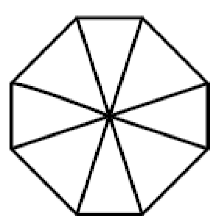
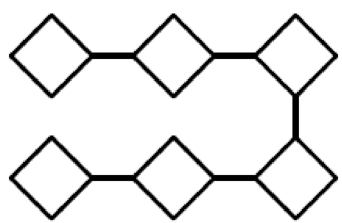
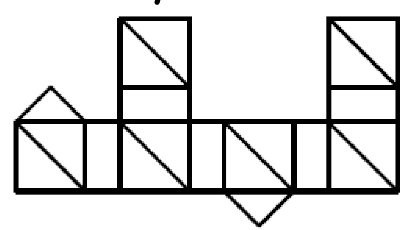
def 1: number of cycles is bounded


good

bad

bad

bad

def 2: removing a bounded number of vertices makes it acyclic


good

good

bad

bad

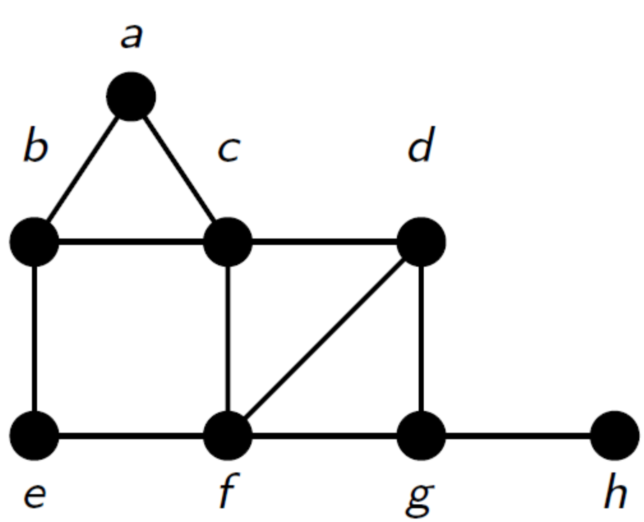
def 3: bounded-size parts connected in a tree-like way

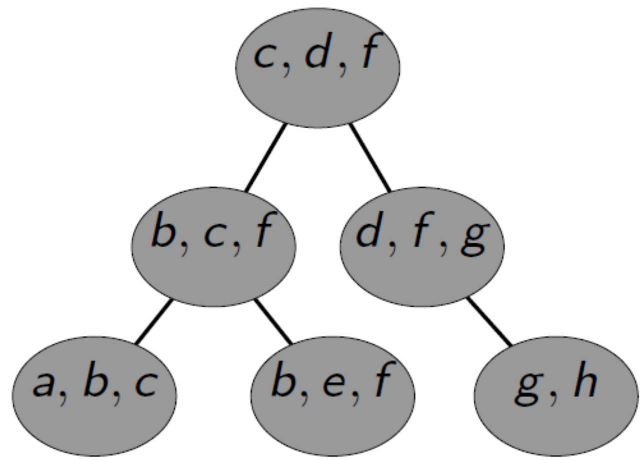

bad

bad

good

good

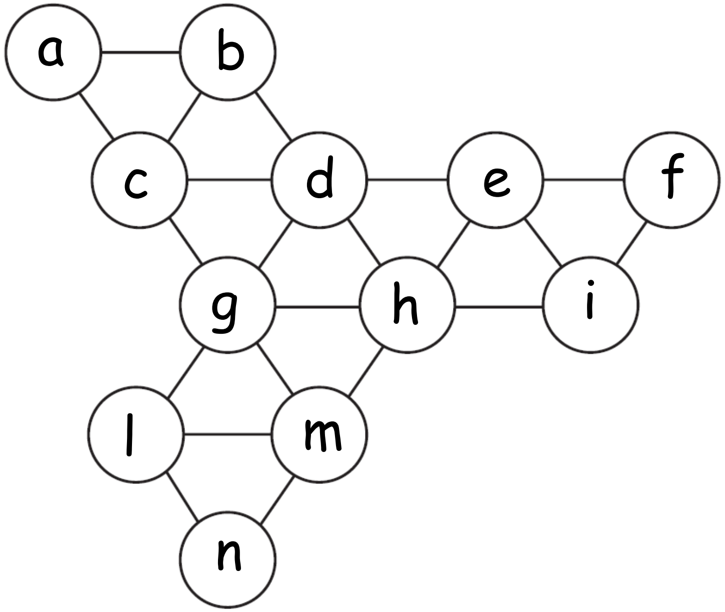
A tree decomposition $(T, \{V_t : t \in T\})$ of a graph $G=(V,E)$ consists of a tree T (on a different node set from G), and a piece $V_t \subseteq V$ associated with each node t of T that satisfies the following three properties:

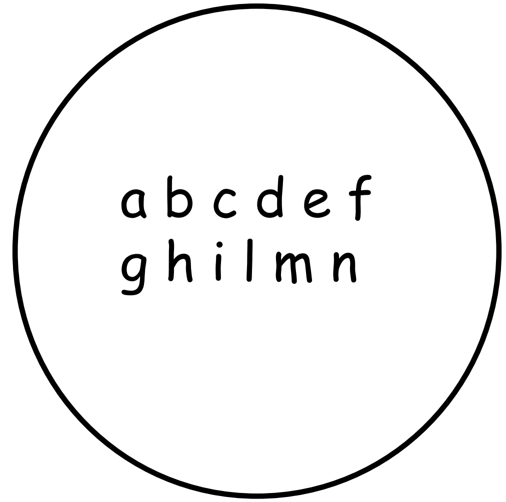
- (Node Coverage): every node of G belongs to at least one piece V_t;
- (Edge Coverage): for every edge e of G, there is some piece V_t containing both endpoints of e;
- (Coherence): Let t_1, t_2 and t_3 be three nodes of T such that t_2 lies on the path from t_1 and t_3. Then, if a node v of G belongs to both V_{t_1} and V_{t_3} it also belongs to V_{t_2}

the width of $(T, \{V_t : t \in T\})$: $\max_t |V_t| - 1$

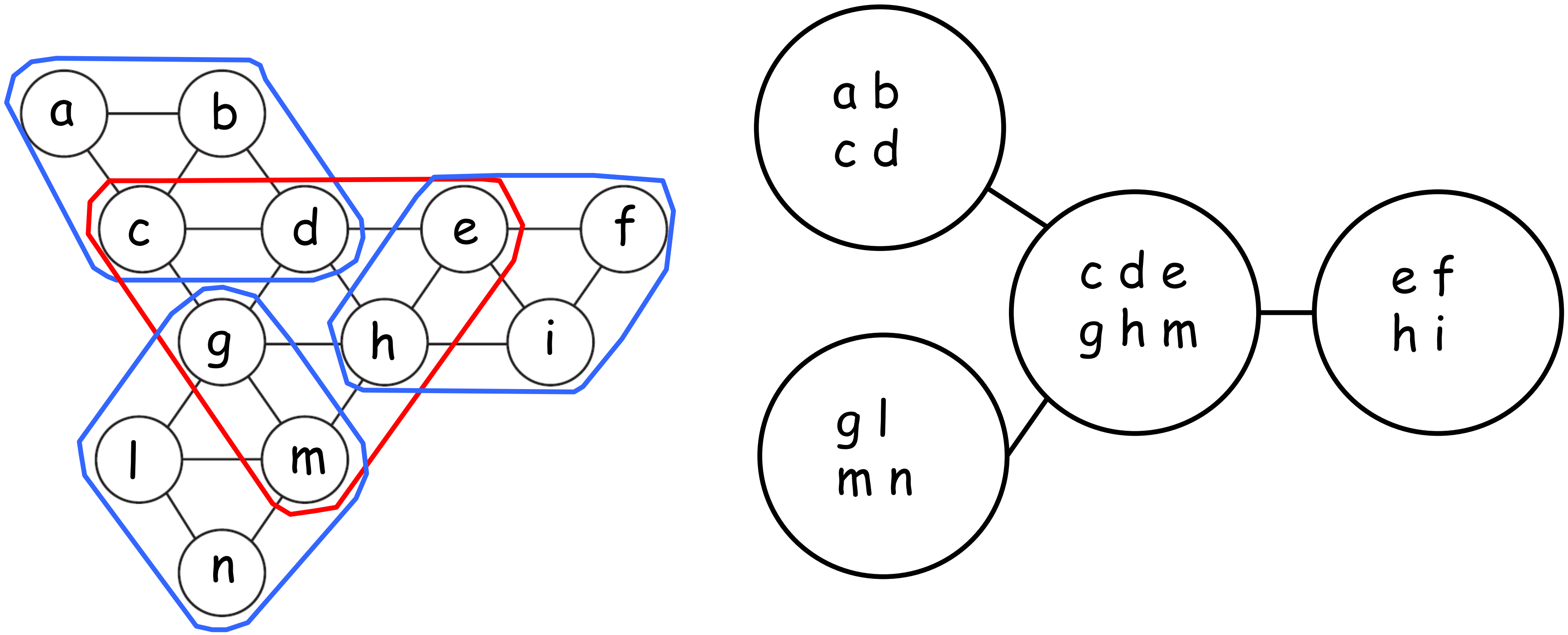




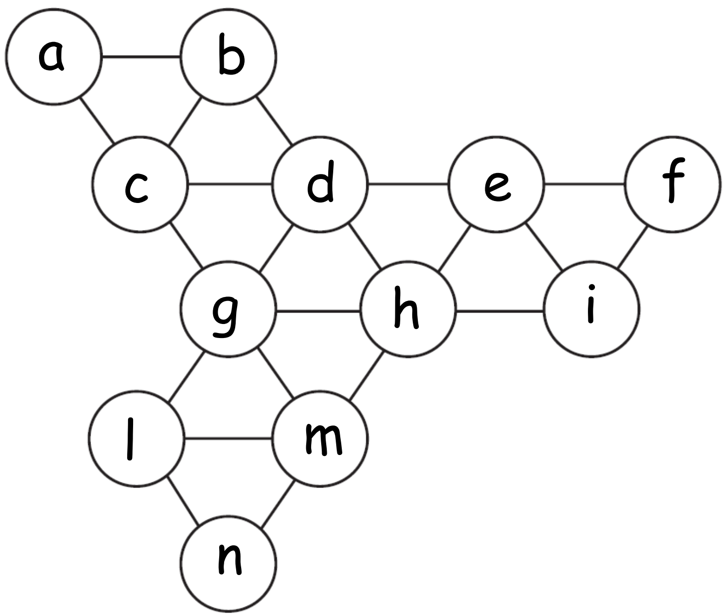




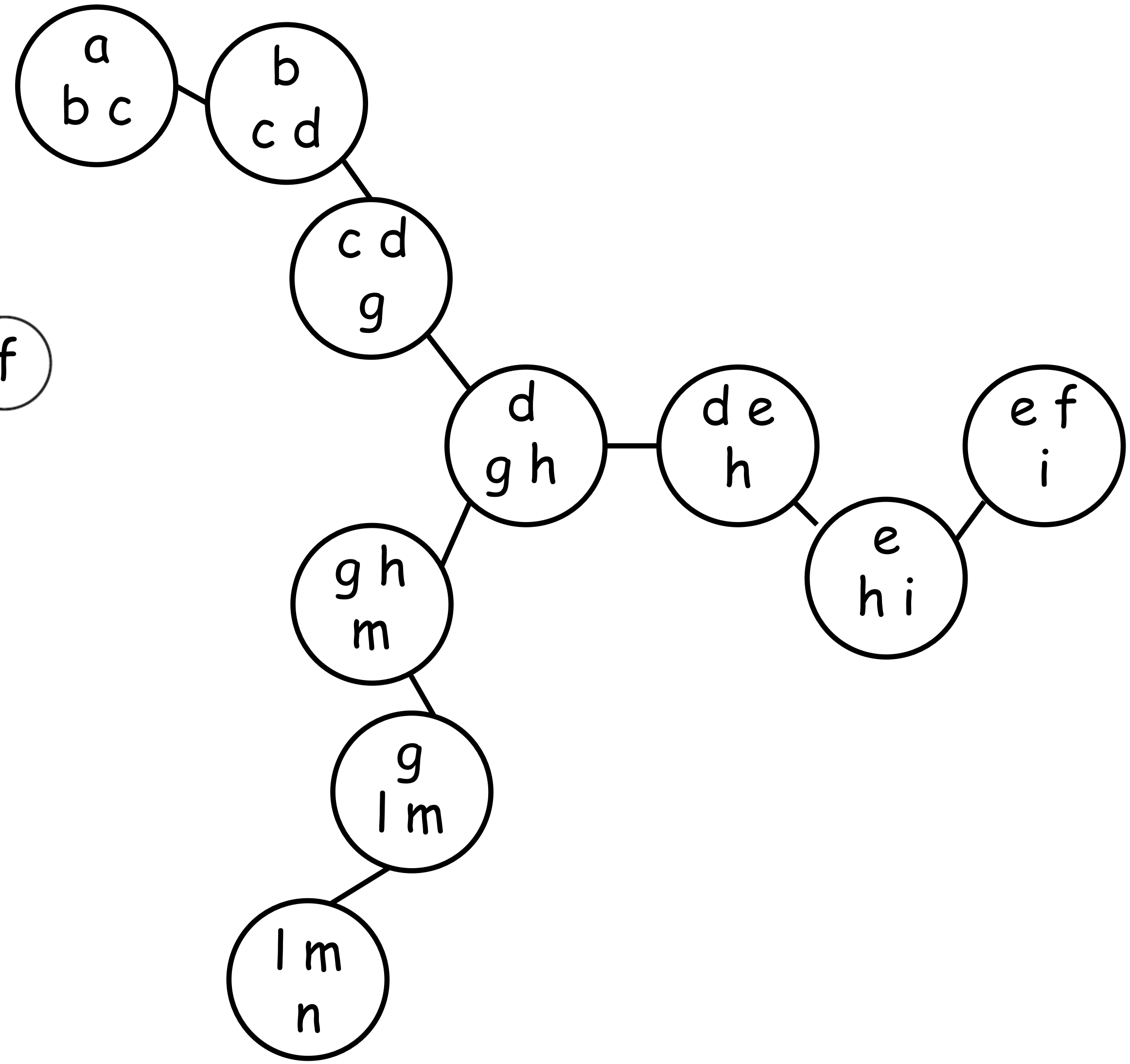
width = 11



width = 5



width = 2

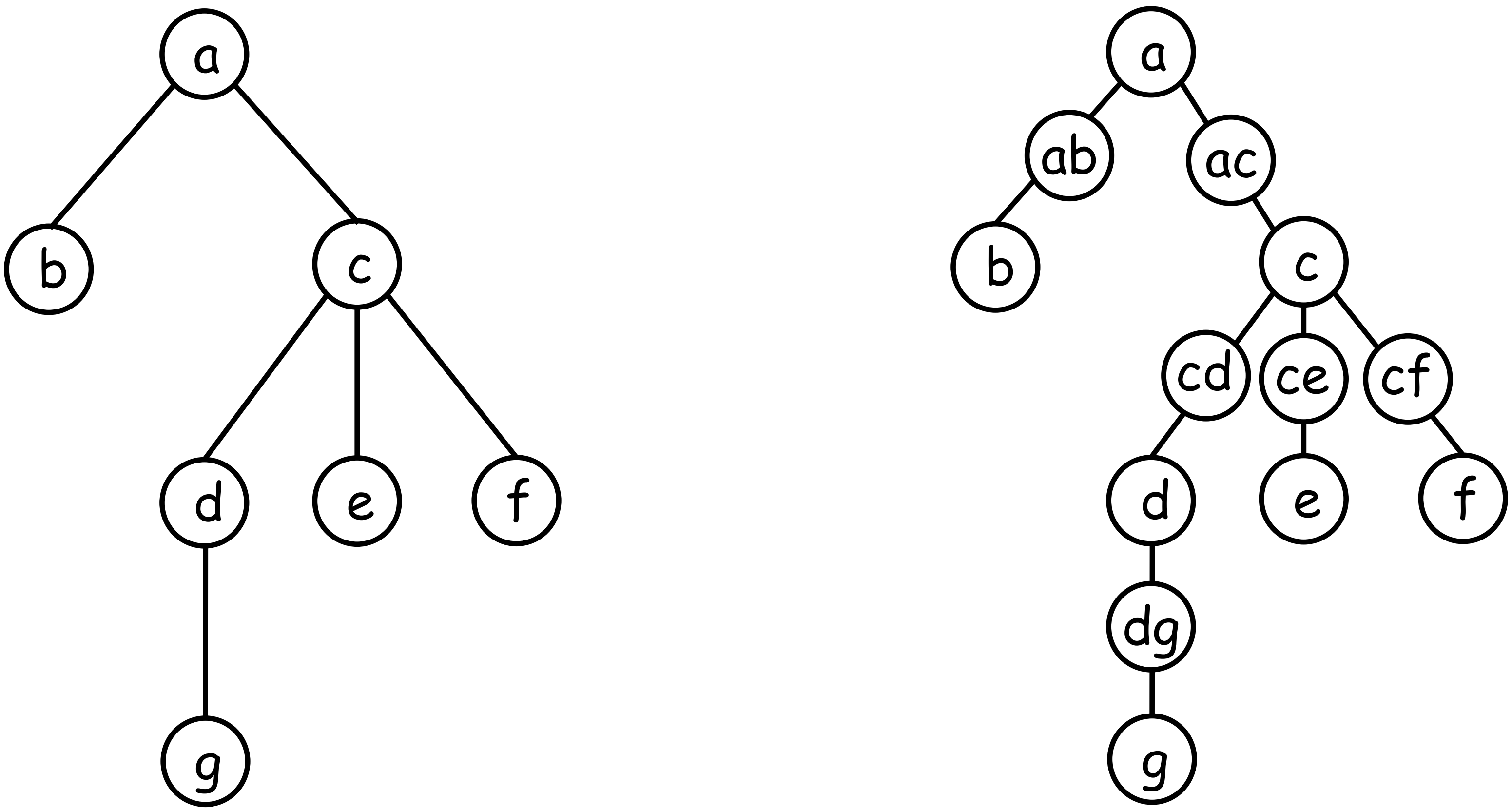


A tree decomposition $(T, \{V_t : t \in T\})$ of a graph $G=(V,E)$ consists of a tree T (on a different node set from G), and a piece $V_t \subseteq V$ associated with each node t of T that satisfies the following three properties:

- (Node Coverage): every node of G belongs to at least one piece V_t;
- (Edge Coverage): for every edge e of G, there is some piece V_t containing both endpoints of e;
- (Coherence): Let t_1, t_2 and t_3 be three nodes of T such that t_2 lies on the path from t_1 and t_3. Then, if a node v of G belongs to both V_{t_1} and V_{t_3} it also belongs to V_{t_2}

the width of $(T, \{V_t : t \in T\})$: $\max_t |V_t| - 1$

the treewidth of G: width of the best tree decomposition of G



the treewidth of a tree is 1

Let T' be a subgraph of T.

$G_{T'}$: subgraph of G induced by the nodes in all pieces associated with nodes of T', that is, the set $\cup_{t \in T'} V_t$.

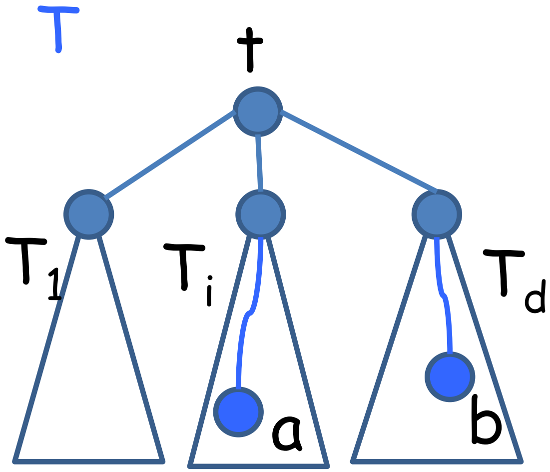
deleting a node t from T

Lemma

Suppose that $T-t$ has components $T_1,\ldots,T_d$. Then the subgraphs

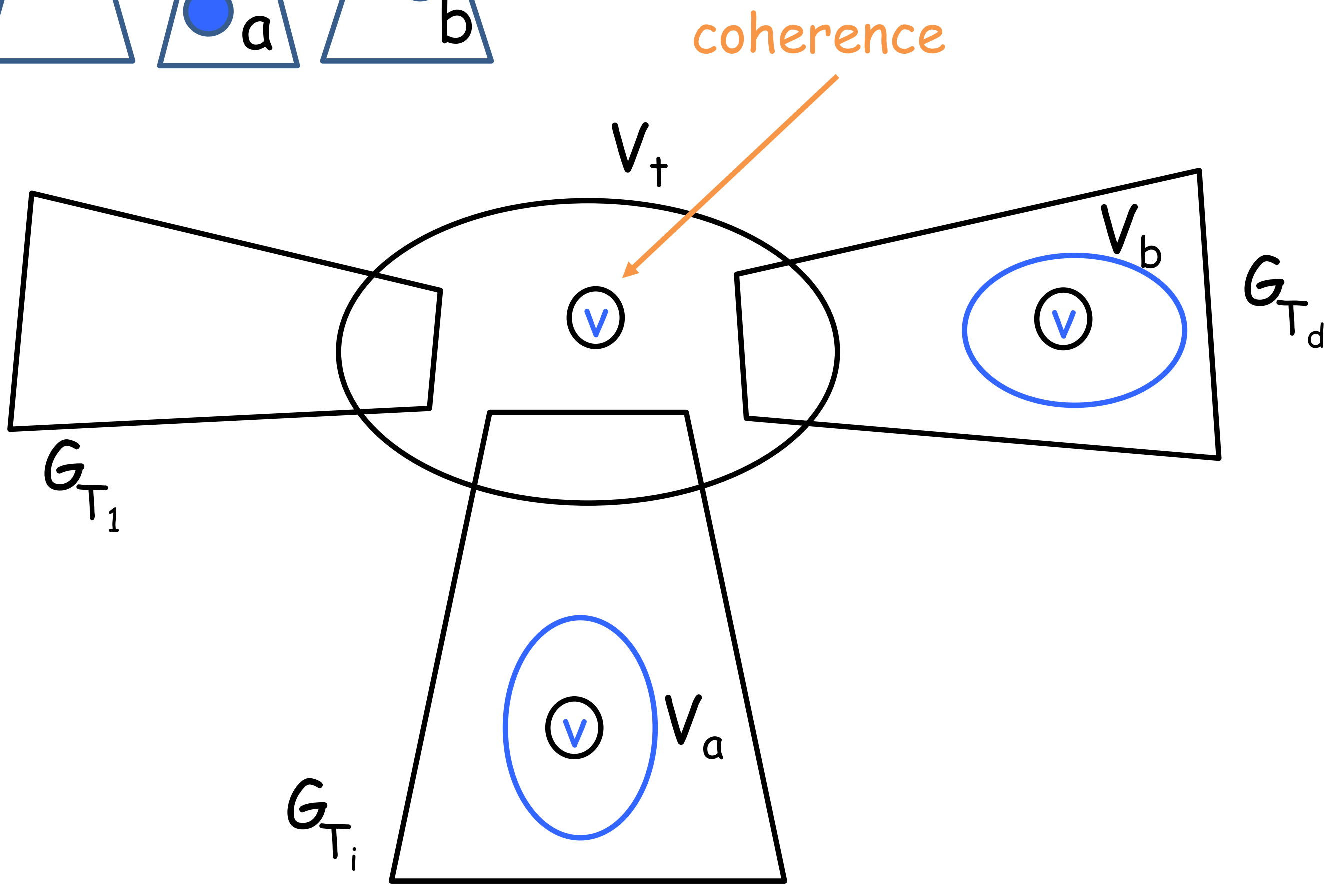
$$G_{T_1} - V_t,\ G_{T_2} - V_t, \ldots,\ G_{T_d} - V_t,$$

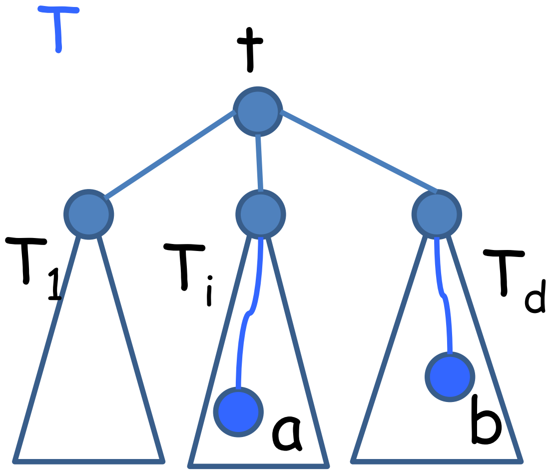
have no nodes in common, and there are no edges between them.



contradiction!

no nodes in common

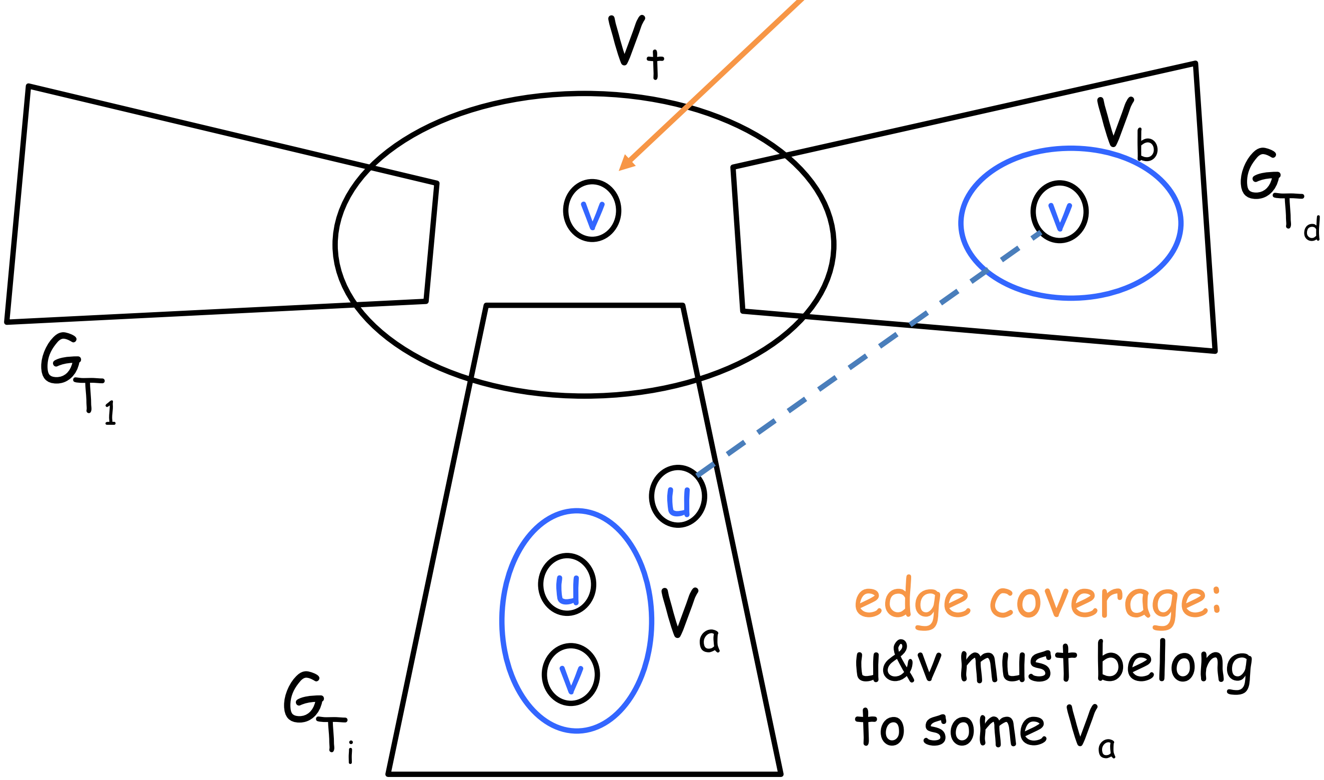




contradiction!

no edges between them

coherence



edge coverage:
u&v must belong to some V_a

Let T' be a subgraph of T.

$G_{T'}$: subgraph of G induced by the nodes in all pieces associated with nodes of T', that is, the set $\cup_{t \in T'} V_t$.

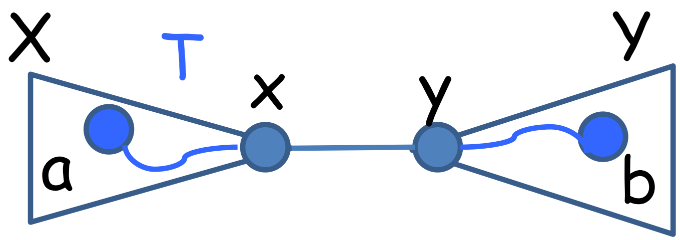
deleting an edge (x,y) from T

Lemma

Let X and Y be the two components of T after the deletion of the edge (x,y). Then the two subgraphs

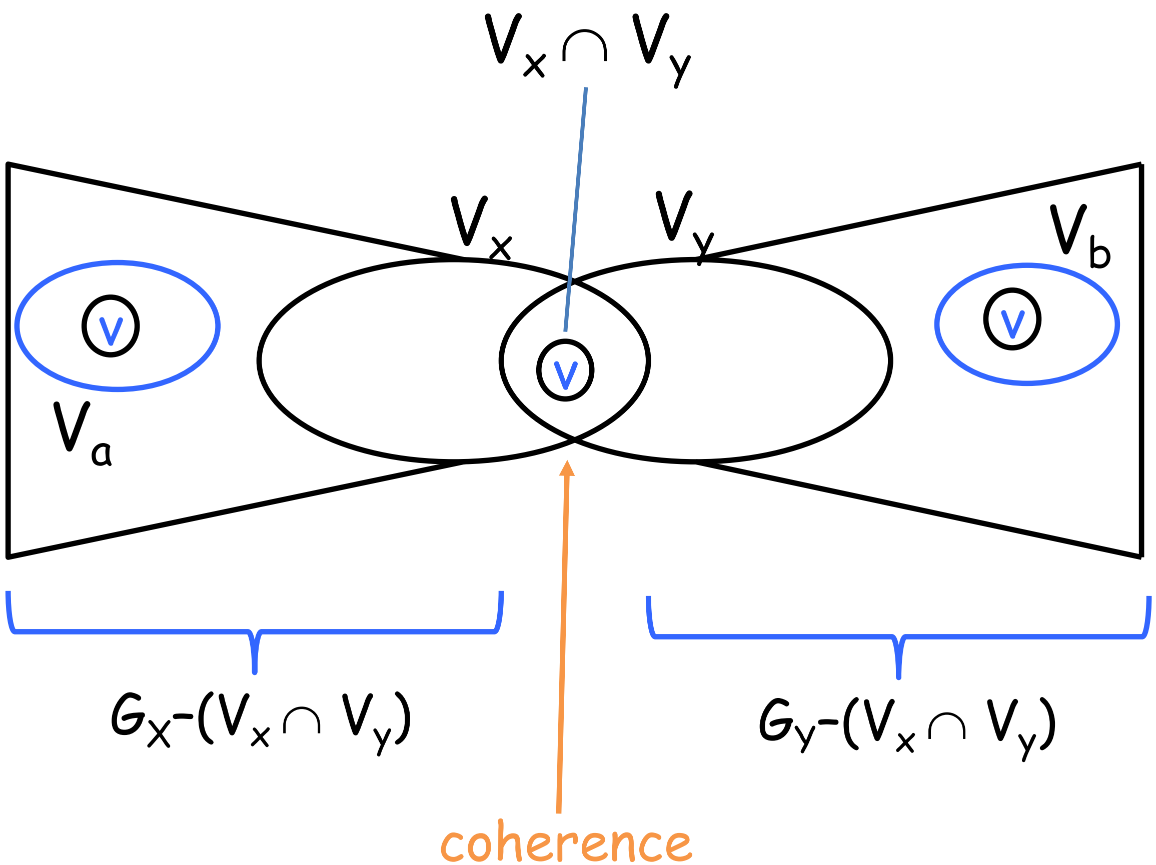
$$G_X-(V_x \cap V_y) \text{ and } G_Y-(V_x \cap V_y)$$

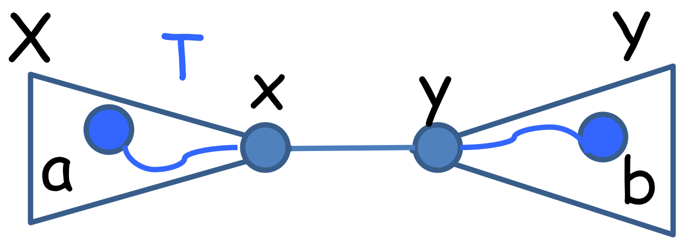
have no nodes in common, and there are no edges between them.



contradiction!

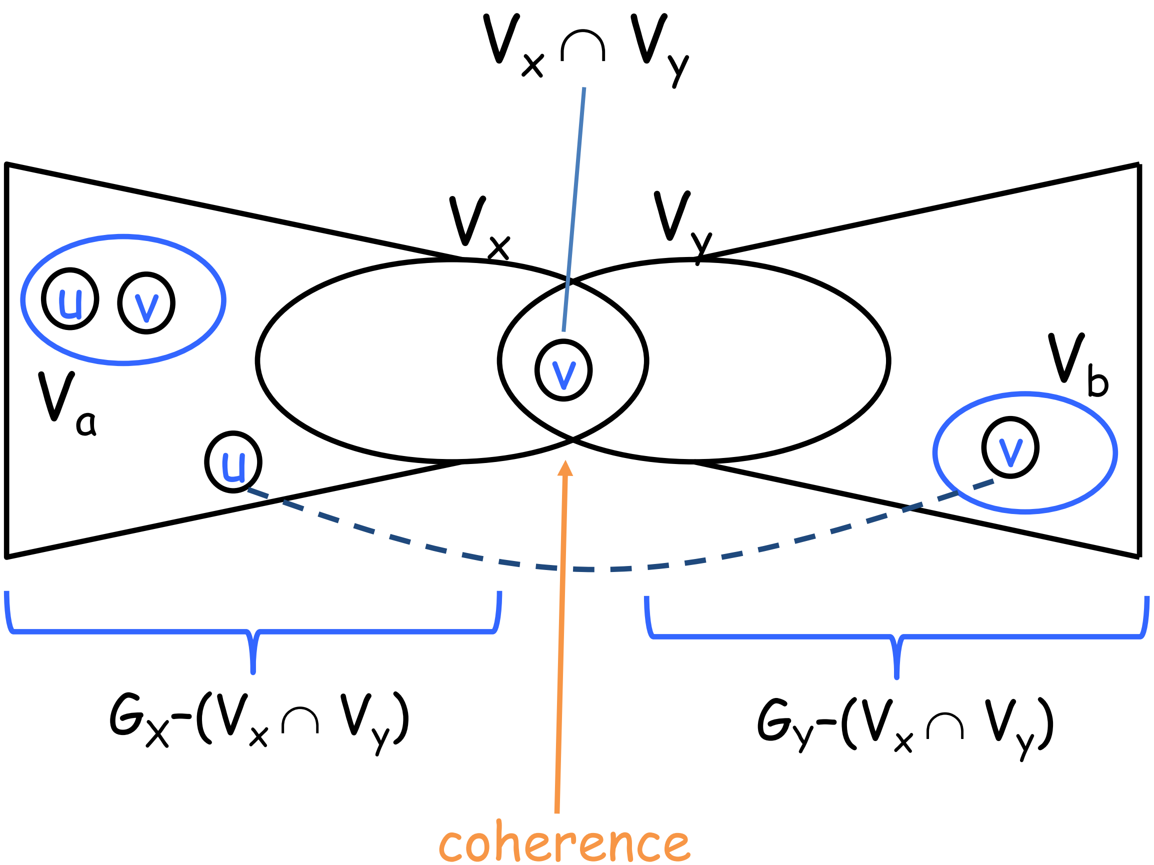
no nodes in common





contradiction!

no nodes in common



A tree decomposition $(T, \{V_t : t \in T\})$ is redundant if there is an edge (x,y) with $V_x \subseteq V_y$.

obtaining a nonredundant tree decomposition:

- whenever a tree decomposition $(T, \{V_t : t \in T\})$ is redundant:
  - contract the edge (x,y) by folding the piece V_x into the piece V_y.

Lemma

Any nonredundant tree decomposition of an n-node graph has at most n pieces.

proof (induction on n.)

n=1 is trivial. Let n>1.

consider a leaf t of T and the corresponding V_t

nonrundancy implies there is at least a node in V_t not in the piece of t's parent (and for coherency in no other piece).

Let U be the set of such nodes

T-t is a nonredundant tree decomposition of G-U with at most

$n-|U| \leq n-1$ pieces $\Rightarrow$ $(T, \{V_t : t \in T\})$ has at most n pieces

■

Dynamic Programming on graph with bounded treewidth w

Solving the weighted Independent Set

defining the subproblems

root T at a node r

for any node t,

- let T_t be the subtree of T rooted at t
- let G_t be the subgraph of G induce by the nodes of all pieces associated with nodes of T_t

subproblems:

for each node t, and each $U \subseteq V_t$:

$f_t(U)$= maximum weight of an independent set S in G_t , subject to the requirement that $S \cap V_t = U$

obs: $f_t(U) = -\infty$ (or undefined) if U is not an IS

number of subproblems:

2^{w+1} for each node t
$2^{w+1}n$ overall for nonredundant tree decomposition

goal:

compute $\max_{U \subseteq V_r} f_r(U)$

$f_t(U)$ = maximum weight of an independent set S in G_t , subject to the requirement that $S \cap V_t = U$

let S be a maximum-weight IS in G_t subject to the requirement that $S \cap V_t = U$, that is $w(S) = f_t(U)$

assume that t has children $t_1, \ldots, t_d$:

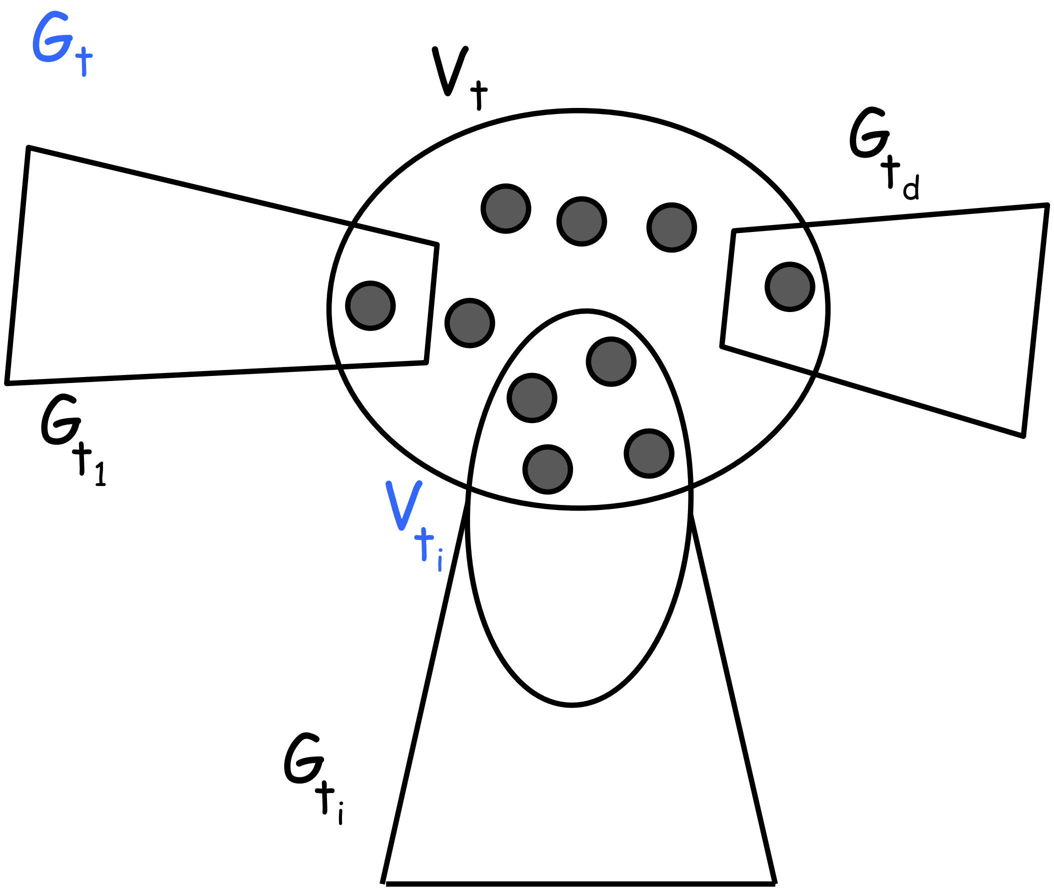
S_i : intersection of S and the nodes of G_{t_i}

Lemma

S_i is a maximum-weight IS of G_{t_i} subject to

$$S_i \cap V_t = U \cap V_{t_i}$$

$f_t(U)$ = maximum weight of an independent set S in G_t, subject to the requirement that $S \cap V_t = U$

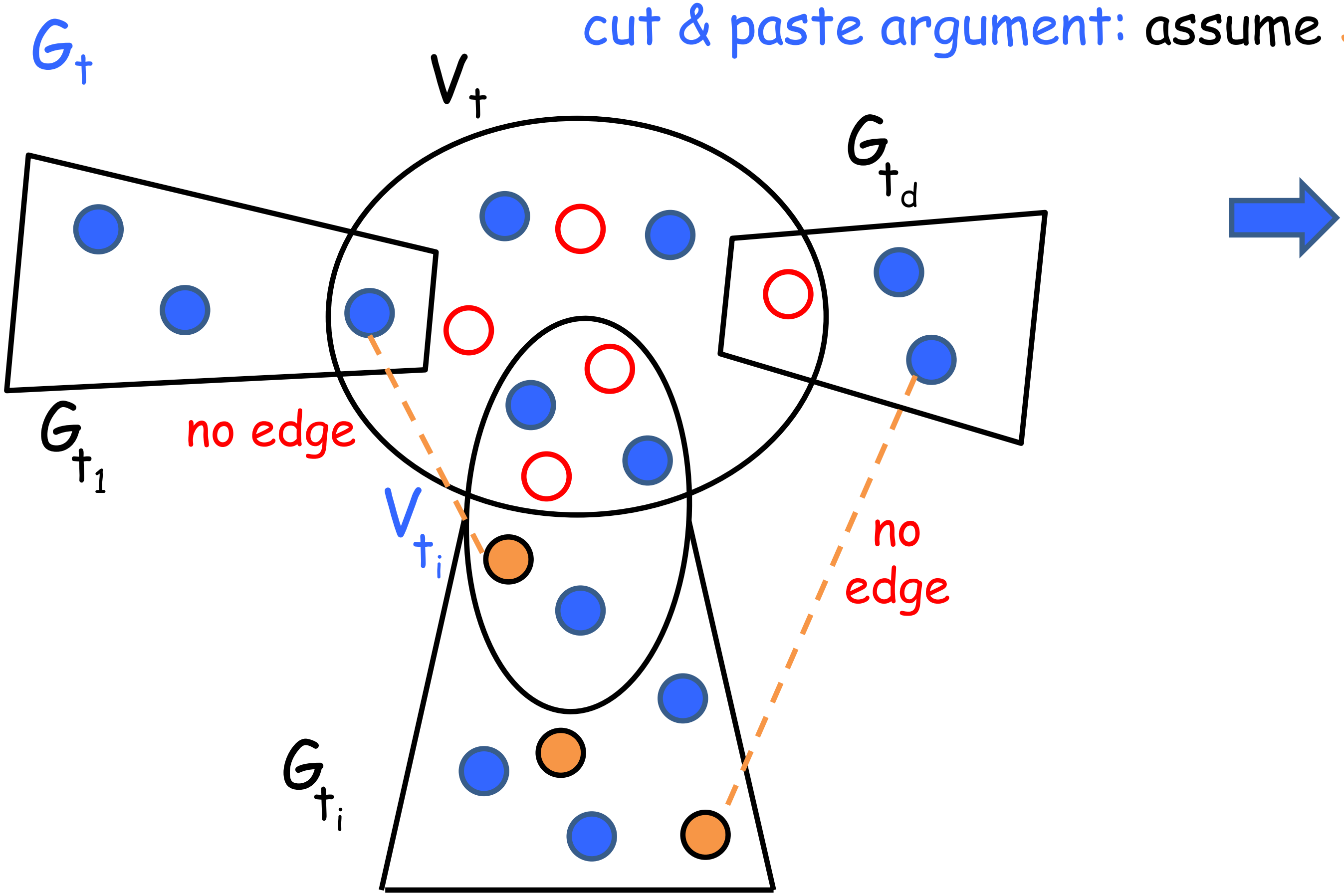


$f_t(U)$ = maximum weight of an independent set S in G_t, subject to the requirement that $S \cap V_t = U$

S_i : intersection of S and the nodes of G_{t_i}

claim: S_i is opt for G_{t_i}, subject to $S_i \cap V_t = U \cap V_{t_i}$

cut & paste argument: assume S'_i better than S_i



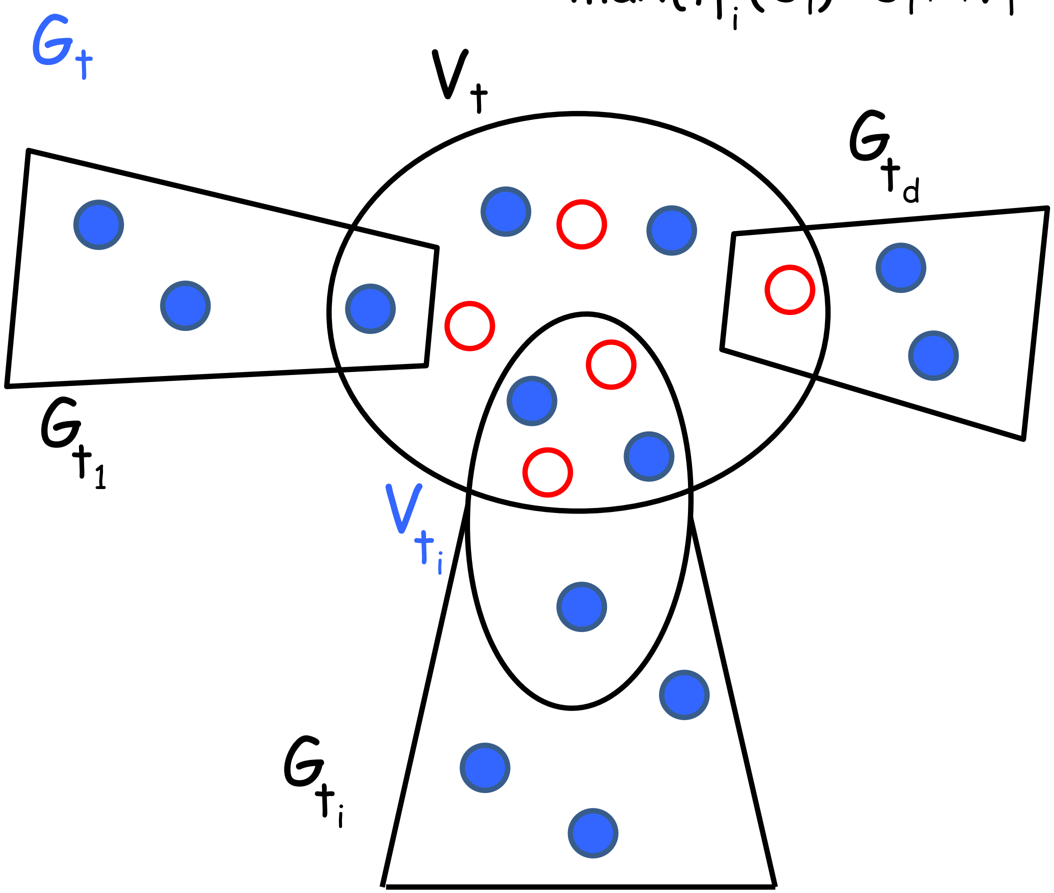
$S' = (S - S_i) \cup S'_i$ is a better IS for G_t

$f_t(U) =$ maximum weight of an independent set S in G_t, subject to the requirement that $S \cap V_t = U$

S_i : intersection of S and the nodes of G_{t_i}

weight of such an optimal S_i :

$\max\{f_{t_i}(U_i) \colon U_i \cap V_t = U \cap V_{t_i} \text{ and } U_i \subseteq V_{t_i} \text{ is an IS}\}$



case: t leaf in T

$U \subseteq V_t$ independent set

$f_t(U) = w(U)$

case: t has children $t_1, \dots, t_d$ in T

$$f_t(U) = w(U) + \sum_{i=1}^{d} \max\{\, f_{t_i}(U_i) - w(U_i \cap U) \;:\; U_i \cap V_t = U \cap V_{t_i} \text{ and } U_i \subseteq V_{t_i} \text{ is an IS} \,\}$$

```
To find a maximum-weight independent set of G,
given a tree decomposition (T, {V_t}) of G:
   Modify the tree decomposition if necessary so it is nonredundant
   Root T at a node r
   For each node t of T in post-order
       If t is a leaf then
            For each independent set U of V_t
                f_t(U) = w(U)
       Else
            For each independent set U of V_t
                f_t(U) is determined by the recurrence
       Endif
    Endfor
    Return max {f_r(U) : U ⊆ V_r is independent}.
```

case: t leaf in T

$U \subseteq V_t$ independent set

$f_t(U) = w(U)$

case: t has children $t_1,\dots,t_d$ in T

$$f_t(U) = w(U) + \sum_{i=1}^{d} \max\{\, f_{t_i}(U_i) - w(U_i \cap U) \;:\; U_i \cap V_t = U \cap V_{t_i} \text{ and } U_i \subseteq V_{t_i} \text{ is an IS } \}$$

time to compute $f_t(U)$:

for each of the d children t_i and each $U_i \subseteq V_{t_i}$

- check in time $O(w)$ if U_i is an IS and is consistent with V_t and U — $O(2^{w+1}\, w\, d)$

there are 2^{w+1} possible U for a node t: $O(4^{w+1}\, w\, d)$

summing over all nodes t:

$\boxed{\text{total running time: } O(4^{w+1}\, w\, n)}$

How to compute a tree-decomposition?

Compute the treewidth of a given graph is NP-hard

There is an algorithm that, given a graph with treewidth w, produce a tree decomposition with width $4w$ in time $O(f(w)\ mn)$