A polynomial-time partitioning algorithm for weighted cactus graphs

Maike Buchin, Leonie Selbach

Ruhr University Bochum

EuroCG 2020

Motivation & Problem

RUB

p-(l, u)-**partition problem**
Given: Vertex-weighted graph $G = (V, E, w)$,
non-negative integers l and u with $l \leq u$,
positive integer p

Motivation & Problem

RUB

p-(l, u)-partition problem
Given: Vertex-weighted graph $G = (V, E, w)$,
non-negative integers l and u with $l \leq u$,
positive integer p

Find a vertex partition into p clusters $V_1, V_2, \ldots, V_p$
such that $l \leq w(V_i) \leq u$ for all $1 \leq i \leq p$.

Motivation & Problem

RUB

p-(l, u)-**partition problem**
Given: Vertex-weighted graph $G = (V, E, w)$,
non-negative integers l and u with $l \leq u$,
positive integer p

connected and pairwise disjoint

Find a vertex partition into p clusters $V_1, V_2, \ldots, V_p$
such that $l \leq w(V_i) \leq u$ for all $1 \leq i \leq p$.

Motivation & Problem

RUB

p-(l, u)-**partition problem**
Given: Vertex-weighted graph $G = (V, E, w)$,
non-negative integers l and u with $l \leq u$,
positive integer p

connected and pairwise disjoint

Find a vertex partition into p clusters $V_1, V_2, \ldots, V_p$
such that $l \leq w(V_i) \leq u$ for all $1 \leq i \leq p$.

$$w(V_i) = \sum_{v \in V_i} w(v)$$

Motivation & Problem

RUB

p-(l,u)-partition problem
Given: Vertex-weighted graph $G = (V, E, w)$,
non-negative integers l and u with $l \leq u$,
positive integer p

connected and pairwise disjoint

Find a vertex partition into p clusters $V_1, V_2, \ldots, V_p$
such that $l \leq w(V_i) \leq u$ for all $1 \leq i \leq p$.

$$= \sum_{v \in V_i} w(v)$$

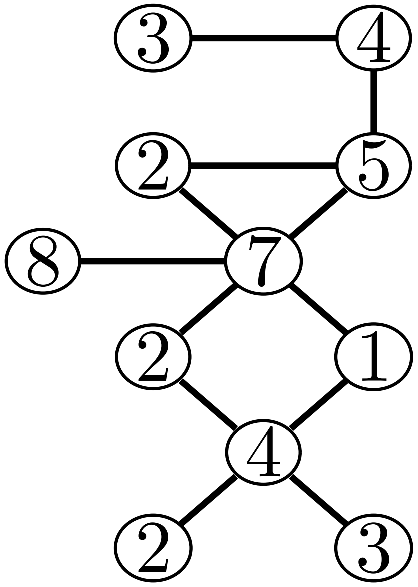
Motivation
Fragmentation of biomedical structures

Motivation & Problem

RUB

p-(l,u)-**partition problem**
Find a (l,u)-partition with exactly p clusters.

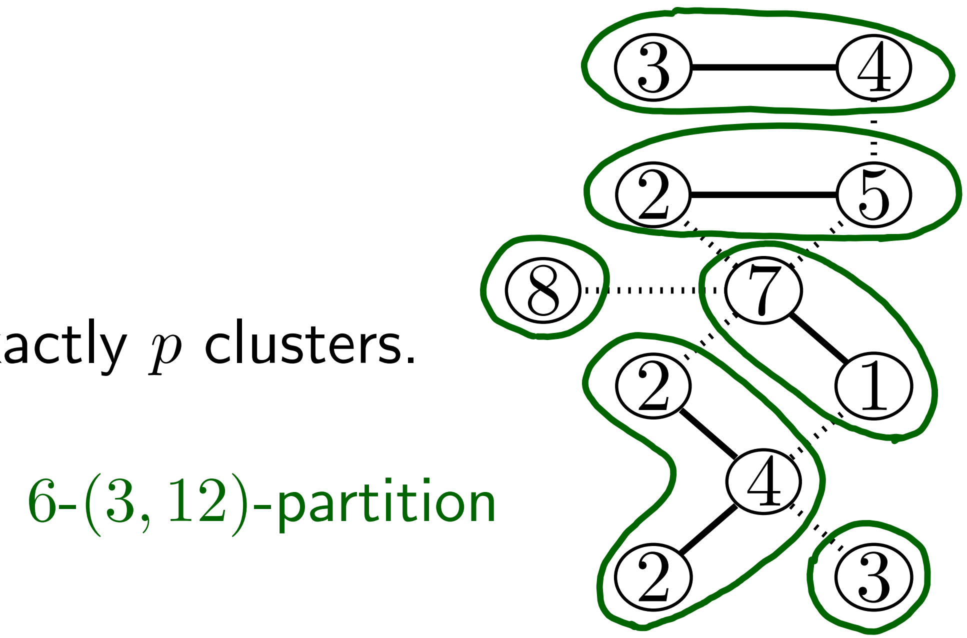
Motivation & Problem



p-(l,u)-**partition problem**
Find a (l,u)-partition with exactly p clusters.

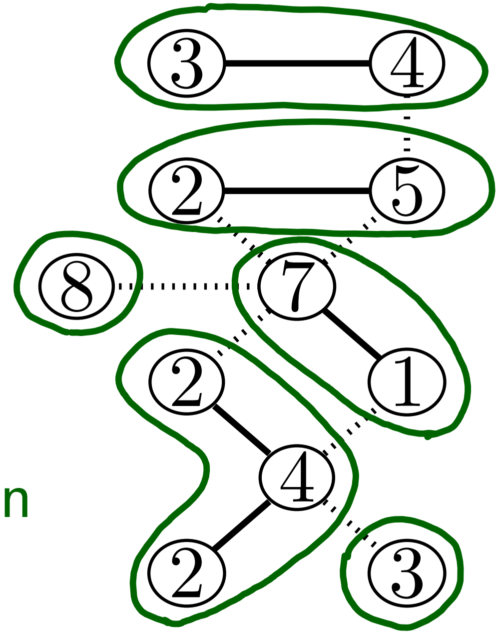
Motivation & Problem

p-(l,u)-**partition problem**
Find a (l,u)-partition with exactly p clusters.



Motivation & Problem

RUB



6-$(3, 12)$-partition

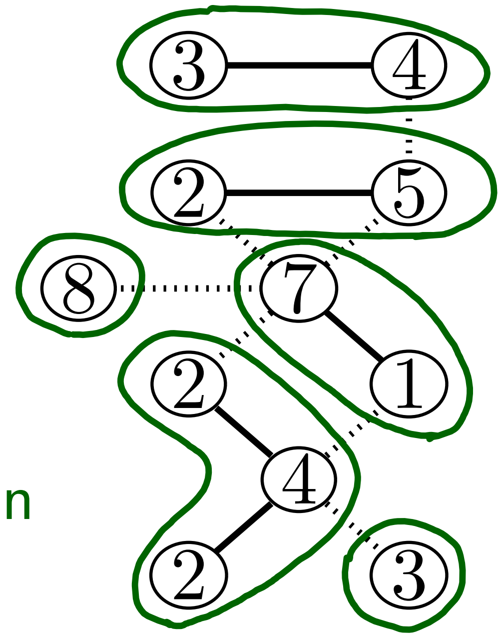
p-(l, u)-**partition problem**
Find a (l, u)-partition with exactly p clusters.

Minimum/maximum-(l, u)**-partition problem**
Find a (l, u)-partition with the minimal resp. maximal number of clusters.

Motivation & Problem

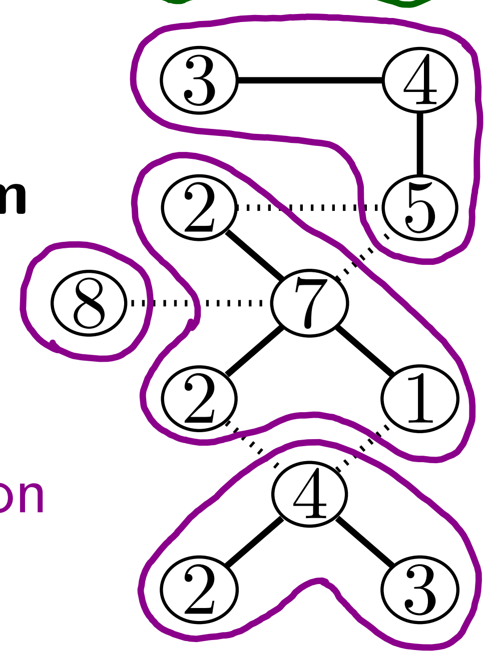
RUB

p-(l,u)-partition problem
Find a (l,u)-partition with exactly p clusters.



6-$(3,12)$-partition

Minimum/maximum-(l,u)-partition problem
Find a (l,u)-partition with the minimal resp. maximal number of clusters.



minimum $(3,12)$-partition

(Related) Results

Related results

- series-parallel graphs **NP-hard**

(Related) Results

RUB

Related results

- series-parallel graphs **NP-hard**
  $\mathcal{O}(u^4 p^2 n)$ $\mathcal{O}(u^4 n)$
- partitial k-trees
  $\mathcal{O}(u^{2(k+1)} p^2 n)$ $\mathcal{O}(u^{2(k+1)} n)$

p-(l, u)-partition problem
min/max-(l, u)-partition problem

(Related) Results

Related results

- series-parallel graphs **NP-hard**
  $\mathcal{O}(u^4p^2n)$ $\mathcal{O}(u^4n)$
- partitial k-trees
  $\mathcal{O}(u^{2(k+1)}p^2n)$ $\mathcal{O}(u^{2(k+1)}n)$
- trees **polynomial** (Decision)
  $\mathcal{O}(p^4n)$ $\mathcal{O}(n^5)$

p-(l, u)-partition problem
min/max-(l, u)-partition problem

(Related) Results

RUB

Related results

- series-parallel graphs **NP-hard**
  $\mathcal{O}(u^4p^2n)$ $\mathcal{O}(u^4n)$
- partitial k-trees
  $\mathcal{O}(u^{2(k+1)}p^2n)$ $\mathcal{O}(u^{2(k+1)}n)$
- trees **polynomial** (Decision)
  $\mathcal{O}(p^4n)$ $\mathcal{O}(n^5)$

p-(l,u)-partition problem
min/max-(l,u)-partition problem

Our results

- cactus graphs **polynomial**
  $\mathcal{O}(p^4n^2)$ $\mathcal{O}(n^6)$

(Related) Results

RUB

Related results

- series-parallel graphs **NP-hard**
  $\mathcal{O}(u^4p^2n)$ $\mathcal{O}(u^4n)$
- partitial k-trees
  $\mathcal{O}(u^{2(k+1)}p^2n)$ $\mathcal{O}(u^{2(k+1)}n)$
- trees **polynomial** (Decision & computation)
  $\mathcal{O}(p^4n)$ $\mathcal{O}(n^5)$

p-(l,u)-partition problem
min/max-(l,u)-partition problem

Our results

- cactus graphs **polynomial** (Decision & Computation)
  $\mathcal{O}(p^4n^2)$ $\mathcal{O}(n^6)$

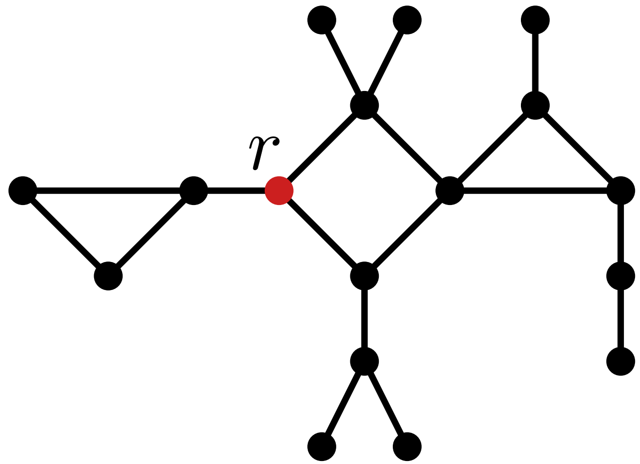
Preprocessing

RUB

DFS on some vertex $r \in G$ and store cycles

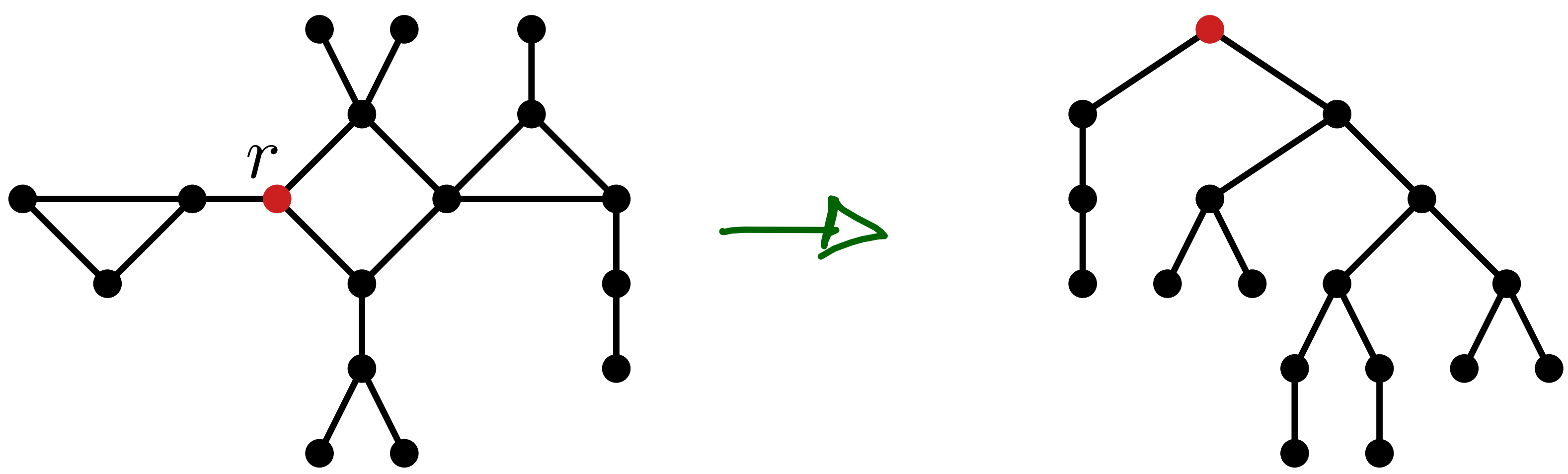
Preprocessing

DFS on some vertex $r \in G$ and store cycles



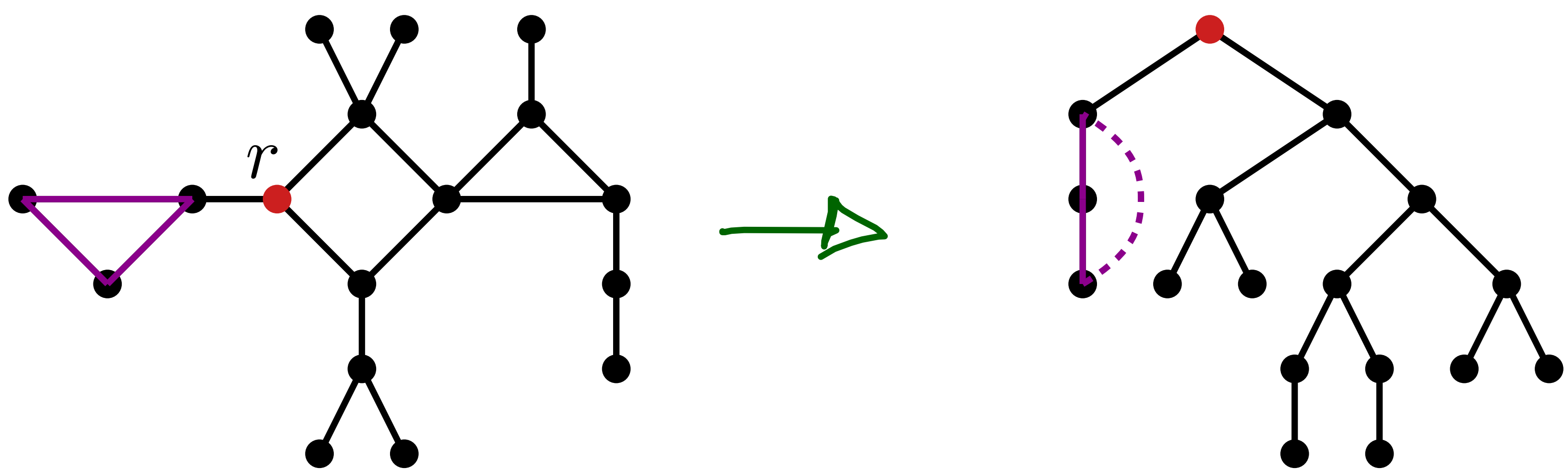
Preprocessing

DFS on some vertex $r \in G$ and store cycles



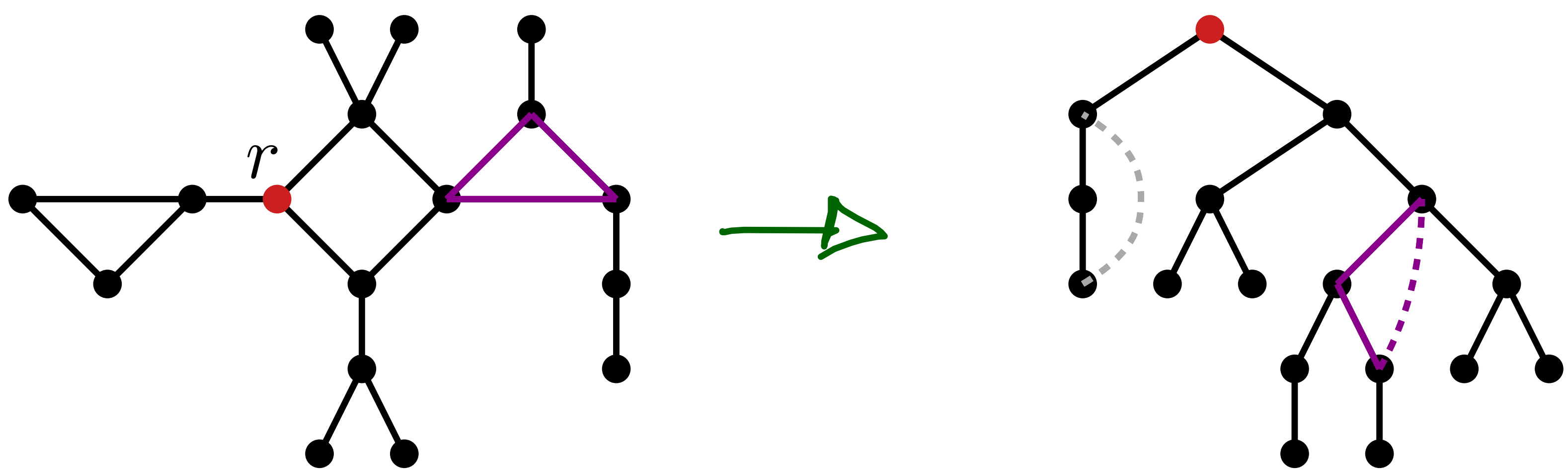
Preprocessing

DFS on some vertex $r \in G$ and store cycles



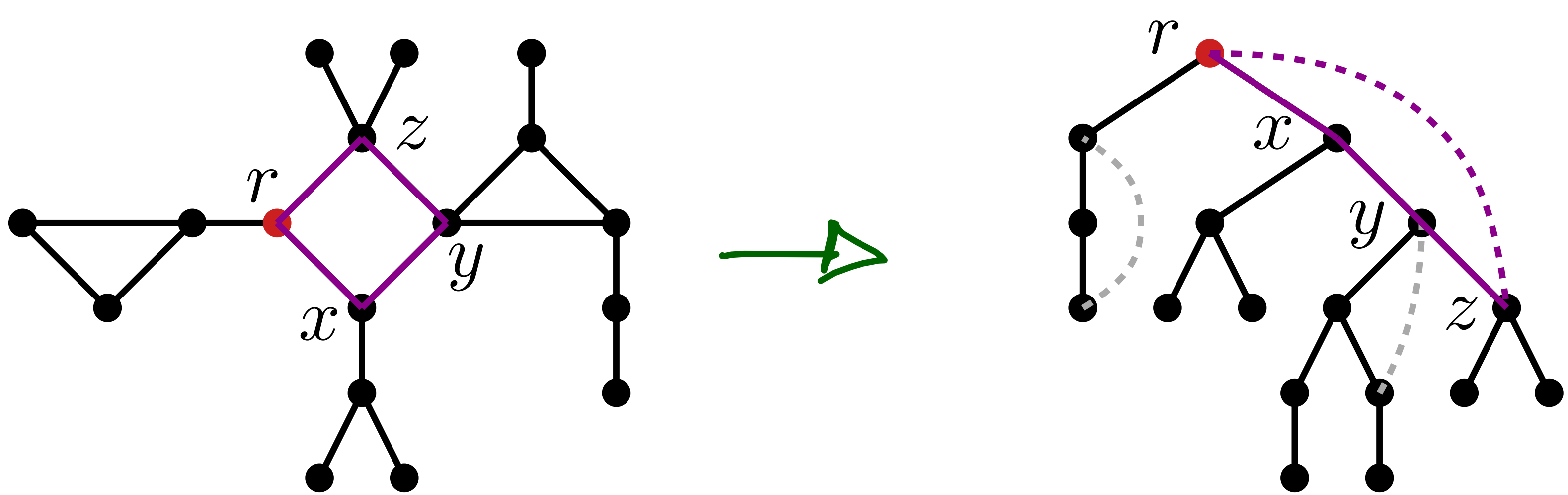
Preprocessing

DFS on some vertex $r \in G$ and store cycles



Preprocessing

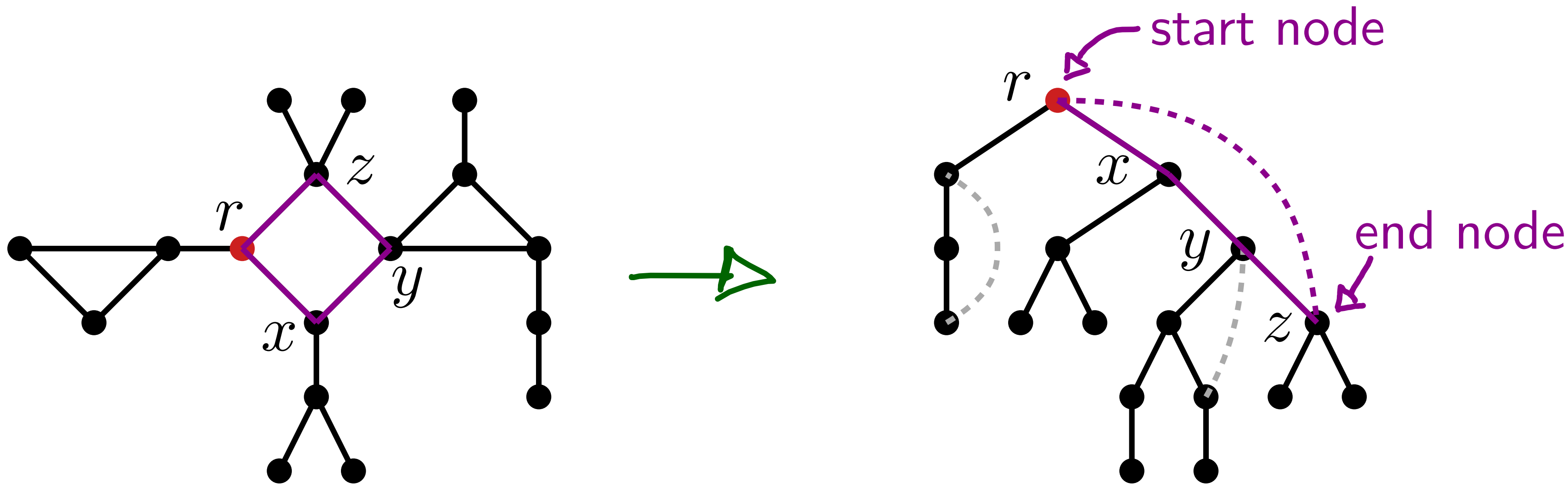
DFS on some vertex $r \in G$ and store cycles



Preprocessing

RUB

DFS on some vertex $r \in G$ and store cycles



$C(r, z) = \langle r, x, y, z \rangle$

Some Definitions

For a partition P of T_v:

$|P|$ = number of clusters in P

C_v = cluster containing the node v

Some Definitions

RUB

subtree rooted in node v

For a partition P of T_v:

$|P|$ = number of clusters in P

C_v = cluster containing the node v

Some Definitions

RUB

For a partition P of T_v: → subtree rooted in node v

$|P|$ = number of clusters in P

C_v = cluster containing the node v

Extendable (l, u)*-partition* of T_v:

- $w(C_v) \leq u$
- $l \leq w(C') \leq u$ for every cluster $C' \neq C_v$

Some Definitions

RUB

Partition Sets

$$S(T_v) = \{(x,k) \mid \exists \text{ extendable } (l,u)\text{-partition } P \text{ of } T_v \text{ such that } |P| = k \wedge w(C_v) = x\}$$

Some Definitions

RUB

Partition Sets

$$S(T_v) = \{(x,k) \mid \exists \text{ extendable } (l,u)\text{-partition } P \text{ of } T_v \text{ such that } |P| = k \wedge w(C_v) = x\}$$

$$|S(T_v)| = \mathcal{O}(up)$$

Some Definitions

RUB

Partition Sets

$$S(T_v) = \{(x,k) \mid \exists \text{ extendable } (l,u)\text{-partition } P \text{ of } T_v \text{ such that } |P| = k \ \wedge \ w(C_v) = x\}$$

Lemma

T_v has p-(l,u)-partition $\Leftrightarrow$
$\exists\ (x,p) \in S(T_v)$ such that $l \leq x \leq u$

Some Definitions

RUB

Partition Sets

$$S(T_v) = \{(x,k) \mid \exists \text{ extendable } (l,u)\text{-partition } P \text{ of } T_v \text{ such that } |P| = k \wedge w(C_v) = x\}$$

Lemma

T_v has p-(l,u)-partition $\Leftrightarrow$
$\exists\ (x,p) \in S(T_v)$ such that $l \leq x \leq u$

Idea

Compute $S(T_r)$ for r being the root of the DFS-Tree

Some Definitions

RUB

Partition Sets

$$S(T_v) = \{(x,k) \mid \exists \text{ extendable } (l,u)\text{-partition } P \text{ of } T_v \text{ such that } |P| = k \wedge w(C_v) = x\}$$

Lemma

T_v has p-(l,u)-partition $\Leftrightarrow$
$\exists\ (x,p) \in S(T_v)$ such that $l \leq x \leq u$

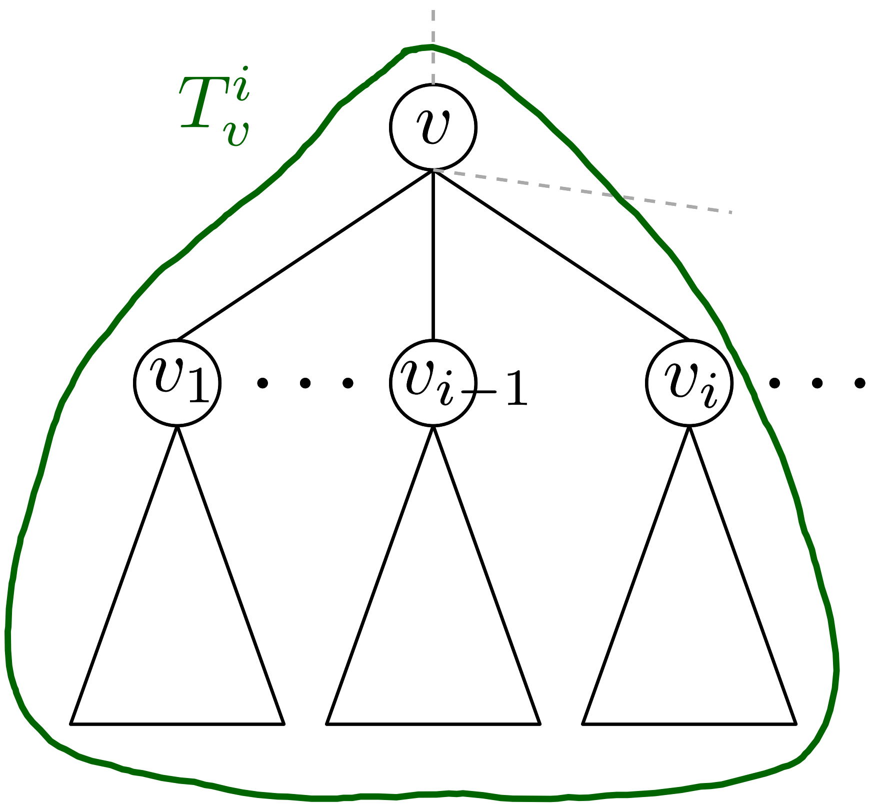
Idea

Compute $S(T_r)$ for r being the root of the DFS-Tree

+ include an efficient procedure for the cycles

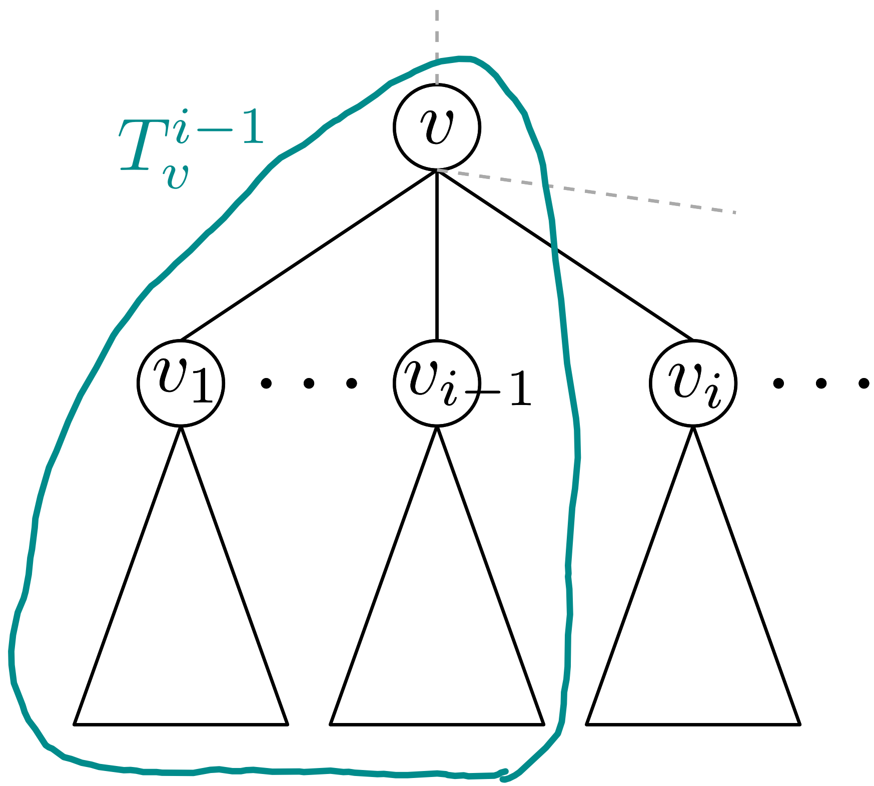
Method

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}



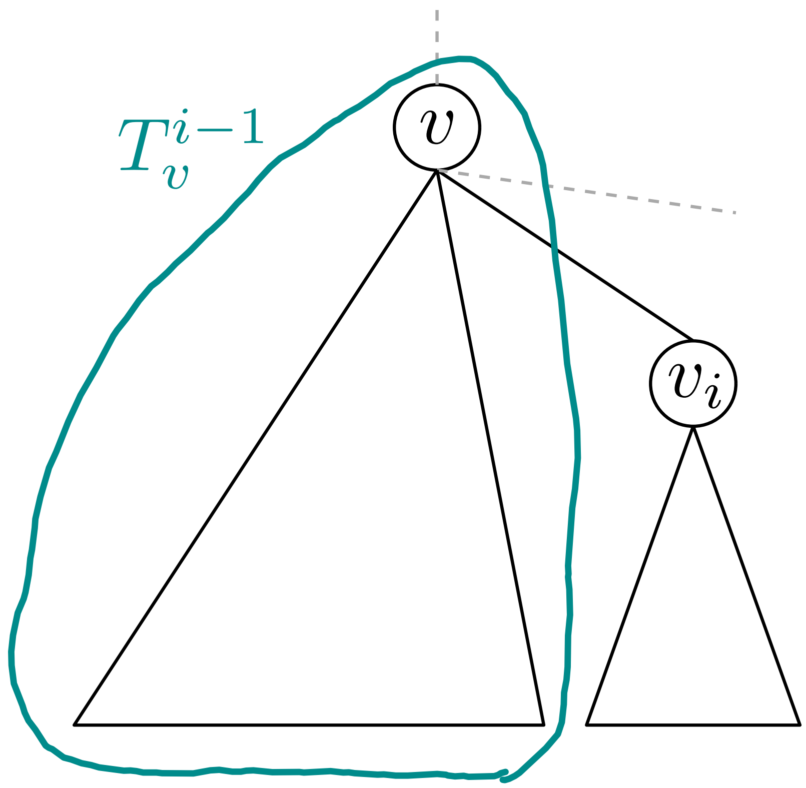
Method

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}



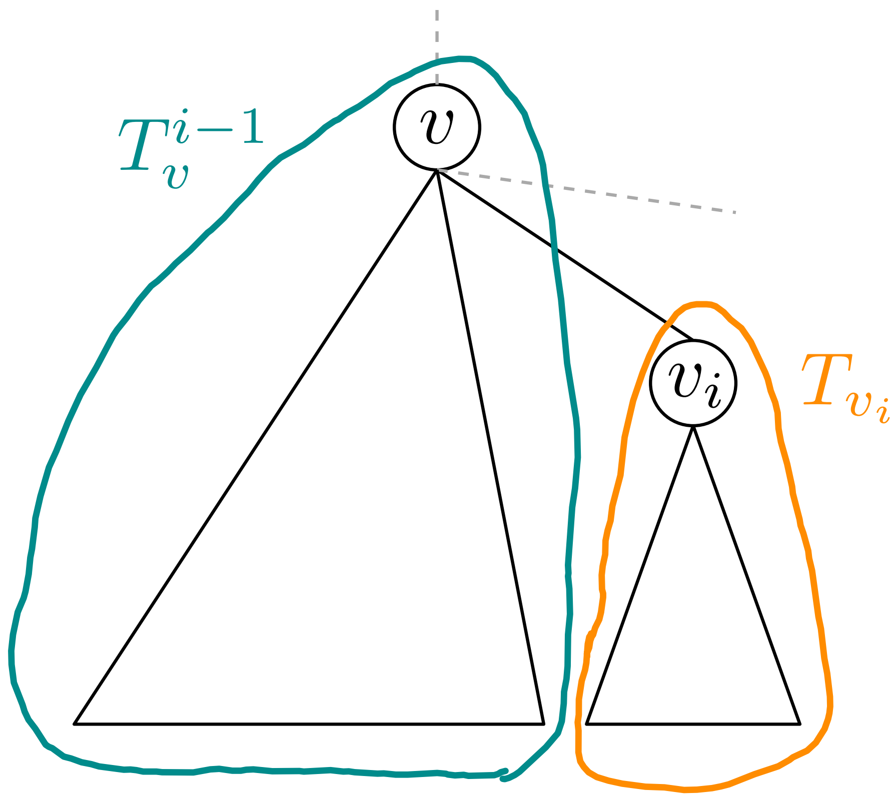
Method

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}



Method

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}

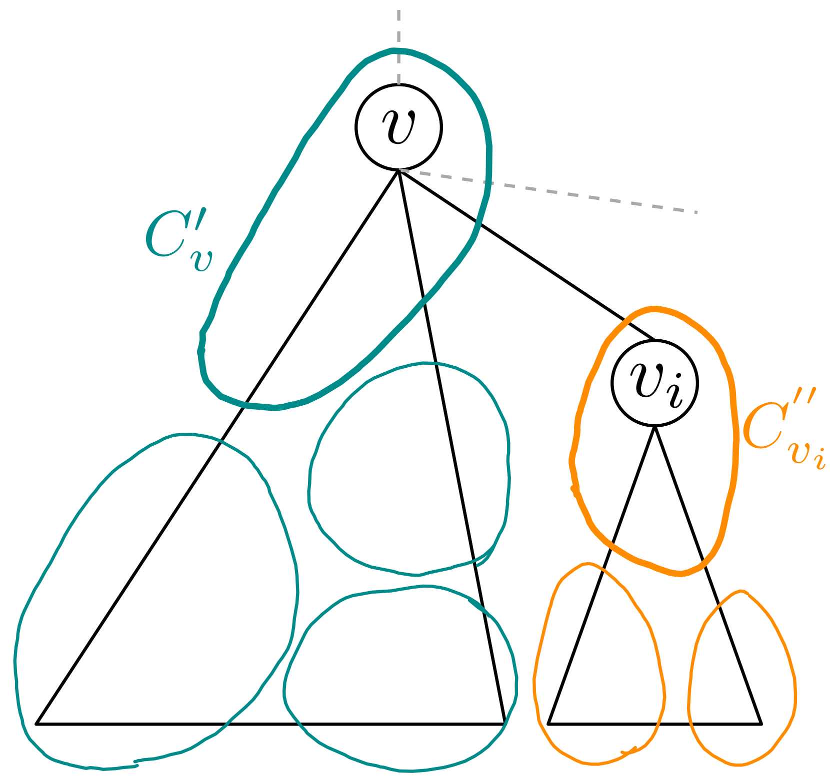


Method

RUB

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}

Given partitions P' of T_v^{i-1} and P'' of T_{v_i}.



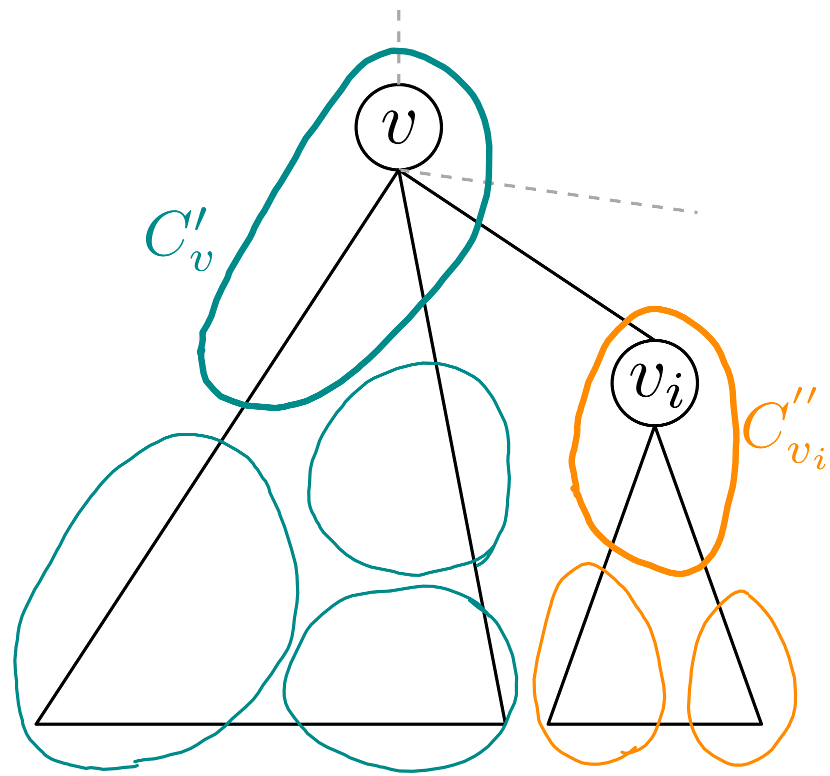
Method

RUB

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}

Given partitions P' of T_v^{i-1} and P'' of T_{v_i}.

as $(x_1, k_1) \in S(T_v^{i-1})$ and $(x_2, k_2) \in S(T_{v_i})$



Method

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}

Option 1: merge

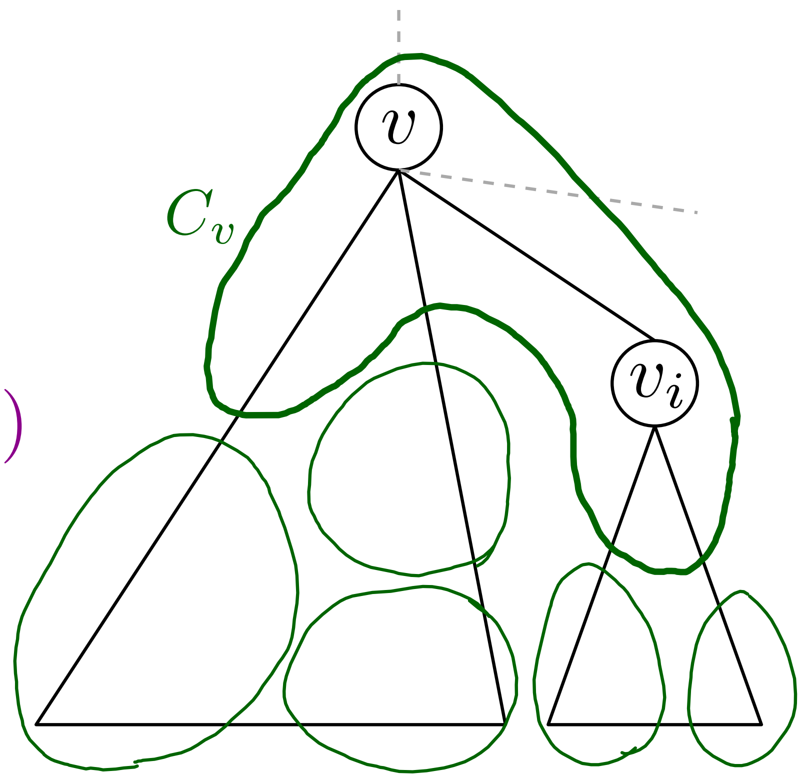
$w(C_v) = w(C'_v) + w(C''_{v_i})$

$|P| = |P'| + |P''| - 1$

$(x_1 + x_2, k_1 + k_2 - 1) \in S(T_v^i)$

if $x_1 + x_2 \leq u$ and

$k_1 + k_2 - 1 \leq p$



Method

RUB

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}

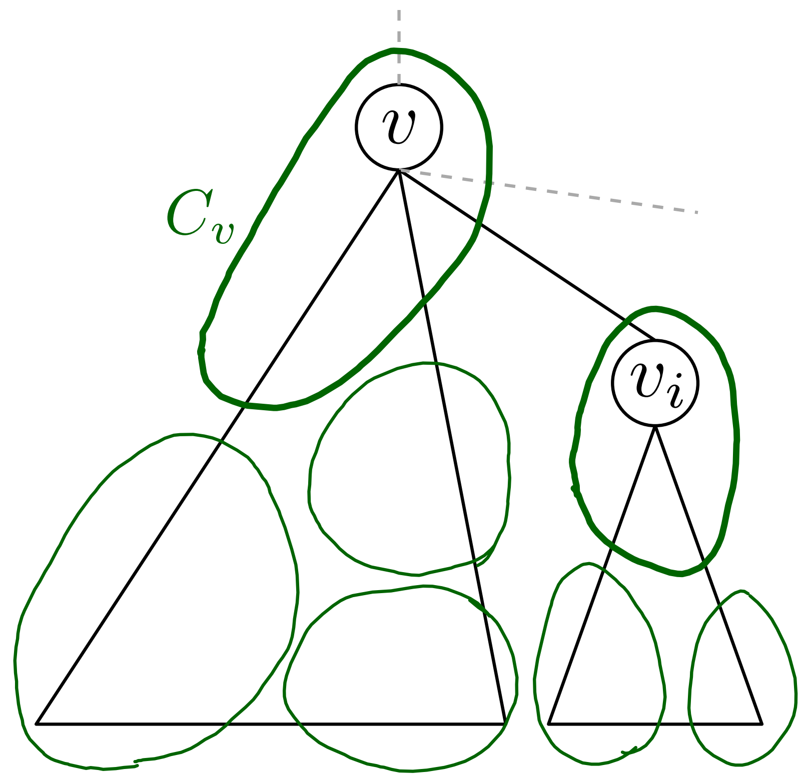
Option 2: don't merge

$w(C_v) = w(C'_v)$

$|P| = |P'| + |P''|$

$(x_1, k_1 + k_2) \in S(T_v^i)$

if $x_2 \geq l$ and $k_1 + k_2 \leq p$



Method

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}

Method

Compute partition P of T_v^i by combining partitions of T_v^{i-1} and T_{v_i}

Compute set $S(T_v^i)$ by combining sets $S(T_v^{i-1})$ and $S(T_{v_i})$

Method

RUB

Compute partition P of T_v^i by
combining partitions of T_v^{i-1} and T_{v_i}

Compute set $S(T_v^i)$
by combining sets $S(T_v^{i-1})$ and $S(T_{v_i})$

↳ $\oplus$-operation $\mathcal{O}(u^2p^2)$

$$
\begin{aligned}
S(T_v^i) =& S(T_v^{i-1}) \oplus S(T_{v_i}) \\
=& \{(x_1 + x_2, k_1 + k_2 - 1) | x_1 + x_2 \leq u, k_1 + k_2 - 1 \leq p\} \\
& \cup \{(x_1, k_1 + k_2) | l \leq x_2, k_1 + k_2 \leq p\}
\end{aligned}
$$

Method

RUB

Compute partition P of T_v^i by
combining partitions of T_v^{i-1} and T_{v_i}

Compute set $S(T_v^i)$
by combining sets $S(T_v^{i-1})$ and $S(T_{v_i})$

↳ $\oplus$-operation

Dynamic approach

$S(T_v^0) = \{(w(v), 1)\}$

$S(T_v^i) = S(T_v^{i-1}) \oplus S(T_{v_i})$ for all edges (v, v_i)

bottom-up

Method

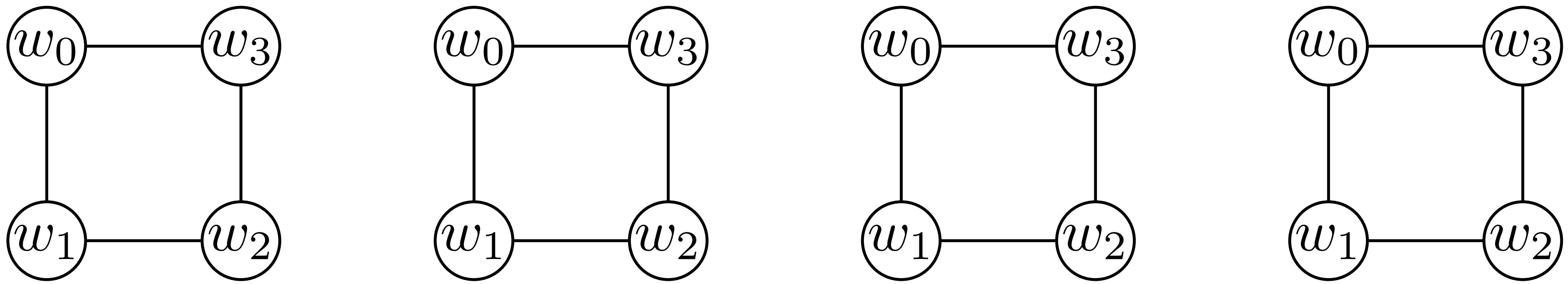
What about cycles?

↳ consider different *configurations*

Method

What about cycles?

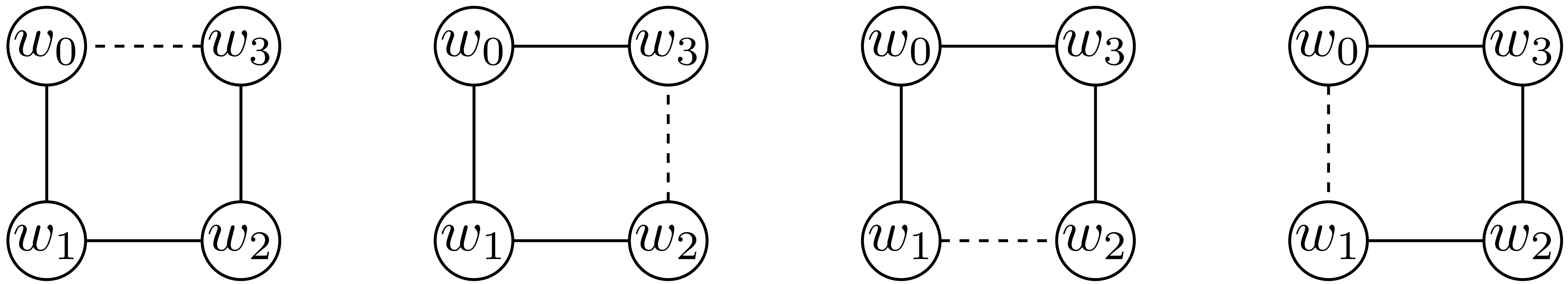
↳ consider different *configurations*



Method

What about cycles?

↳ consider different *configurations*

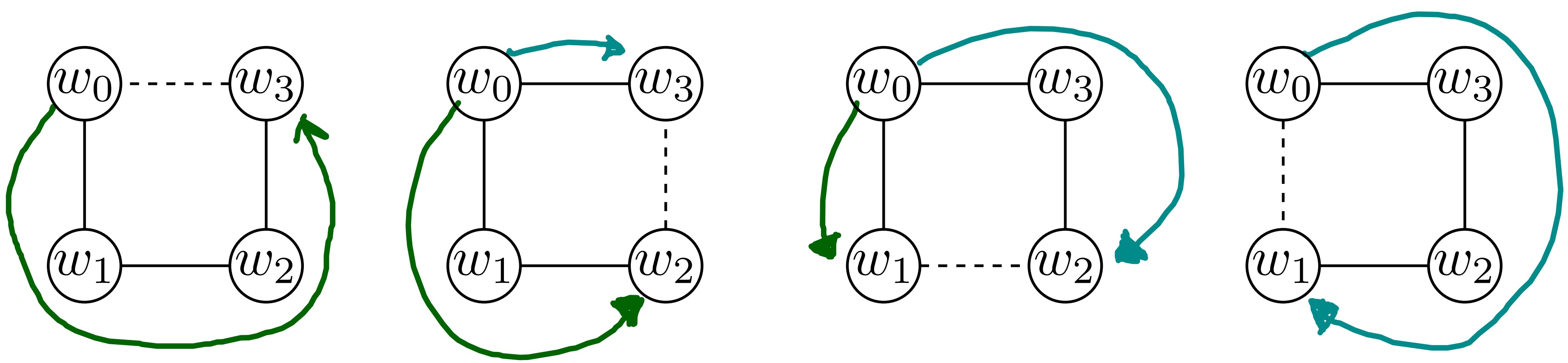


Method

RUB

What about cycles?

↳ consider different *configurations*

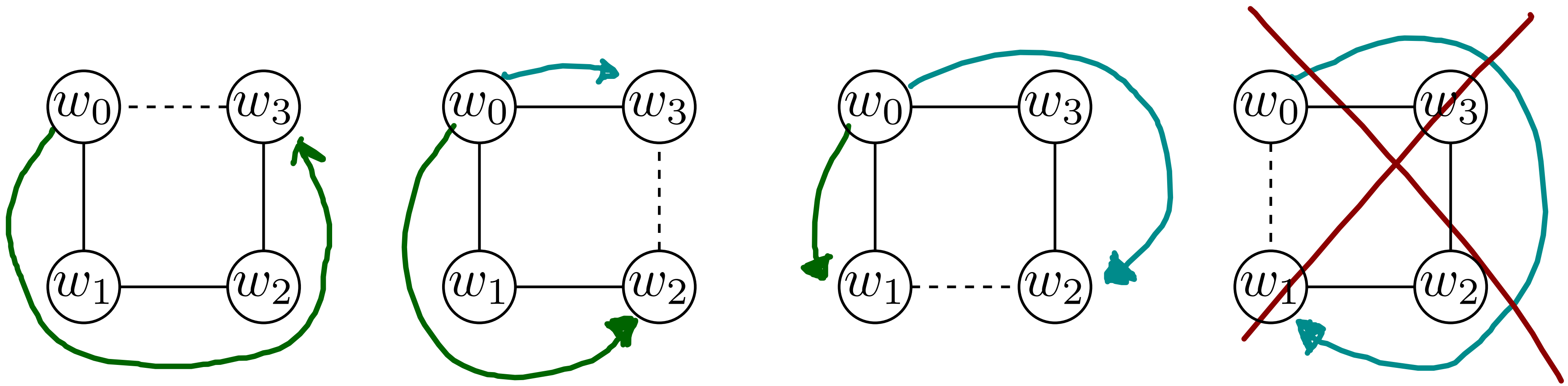


Method

RUB

What about cycles?

consider different *configurations*



Method

What about cycles in the graph?

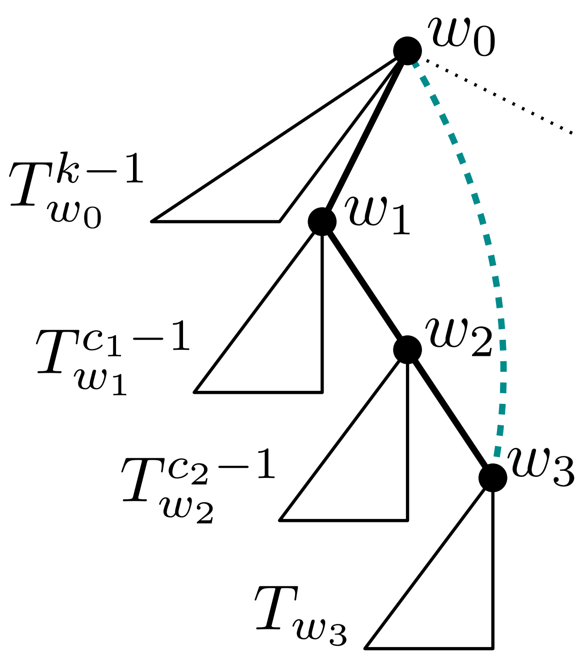
↳ consider different *configurations*

Method

RUB

What about cycles in the graph?

↳ consider different *configurations*

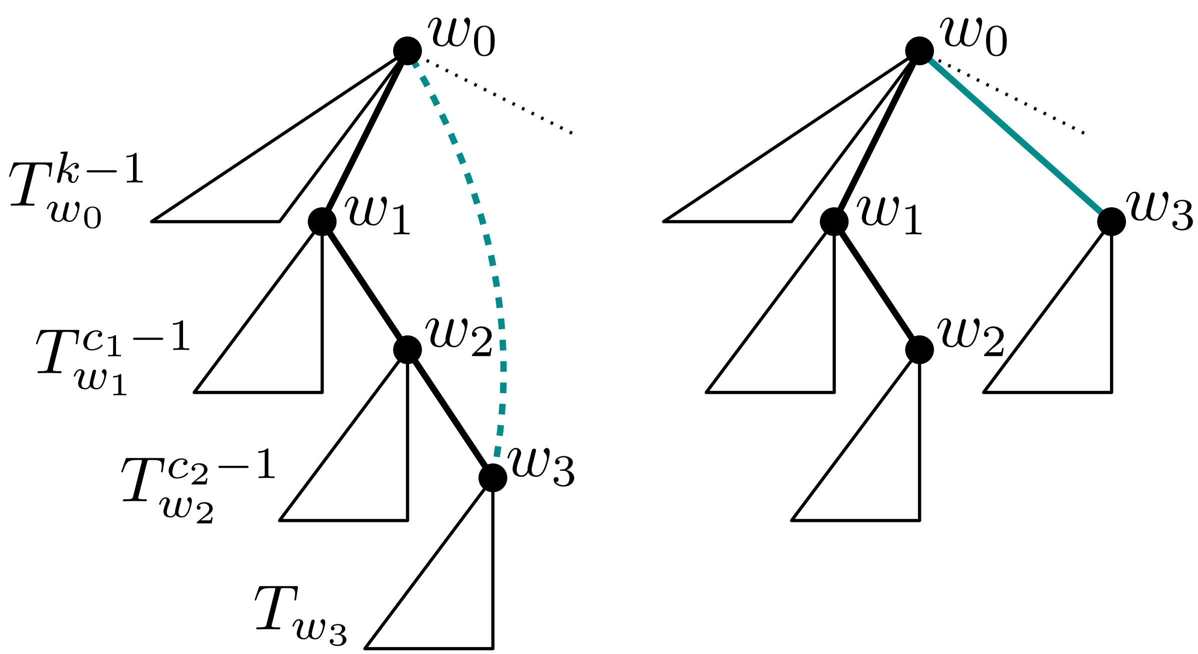


Method

RUB

What about cycles in the graph?

↳ consider different *configurations*

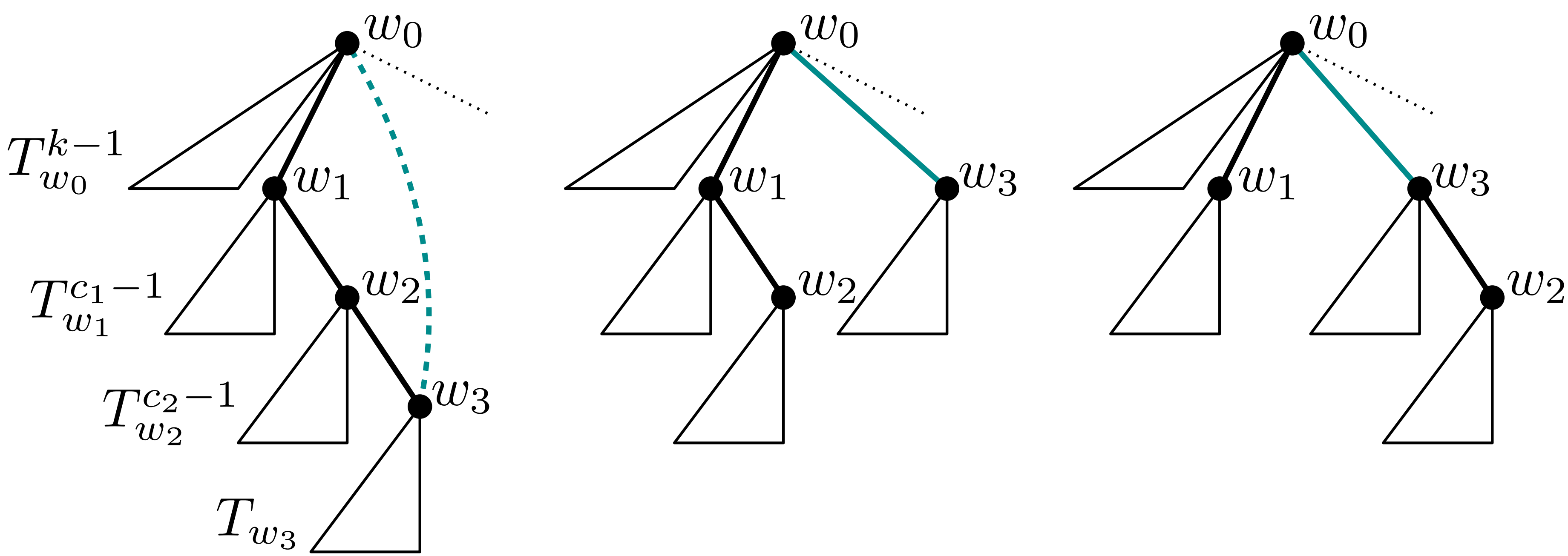


Method

RUB

What about cycles in the graph?

↳ consider different *configurations*



RUB

Computation of the partition sets

for nodes in some cycle $C(w_0, w_{m-1}) = \langle w_0, w_1, \ldots, w_{m-1} \rangle$
in different configurations j

$$S_j(T_{w_{m-1}}) = \begin{cases} S(T_{w_{m-1}}) & j = 1, 2 \\ S(T_{w_{m-1}}) \oplus S_j(T_{w_{m-2}}) & \text{otherwise} \end{cases}$$

$$S_j(T_{w_i}) = \begin{cases} S(T_{w_i}^{c_i - 1}) \oplus S_j(T_{w_{i+1}}) & j < m - i \\ S(T_{w_i}^{c_i - 1}) \oplus S_j(T_{w_{i-1}}) & j > m - i + 1 \\ S(T_{w_i}^{c_i - 1}) & \text{otherwise} \end{cases}$$

$$S_j(T_{w_0}) = \begin{cases} S(T_{w_0}^{k-1}) \oplus S_j(T_{w_1}) & j = 1 \\ \big(S(T_{w_0}^{k-1}) \oplus S_j(T_{w_1})\big) \oplus S_j(T_{w_{m-1}}) & \text{otherwise} \end{cases}$$

RUB

Computation of the partition sets

for nodes in some cycle $C(w_0, w_{m-1}) = \langle w_0, w_1, \ldots, w_{m-1} \rangle$
in different configurations j

$$S_j(T_{w_{m-1}}) = \begin{cases} S(T_{w_{m-1}}) & j = 1, 2 \\ S(T_{w_{m-1}}) \oplus S_j(T_{w_{m-2}}) & \text{otherwise} \end{cases}$$

$$S_j(T_{w_i}) = \begin{cases} S(T_{w_i}^{c_i - 1}) \oplus S_j(T_{w_{i+1}}) & j < m - i \\ S(T_{w_i}^{c_i - 1}) \oplus S_j(T_{w_{i-1}}) & j > m - i + 1 \\ S(T_{w_i}^{c_i - 1}) & \text{otherwise} \end{cases}$$

$$S_j(T_{w_0}) = \begin{cases} S(T_{w_0}^{k-1}) \oplus S_j(T_{w_1}) & j = 1 \\ \left(S(T_{w_0}^{k-1}) \oplus S_j(T_{w_1})\right) \oplus S_j(T_{w_{m-1}}) & \text{otherwise} \end{cases}$$

$$S(T_{w_0}^k) = \bigcup_{j=1}^{m-1} S_j(T_{w_0}^k)$$

Results

Theorem

Given a weighted cactus graph G, a positive integer p and two non-negative integers l and u (with $l \leq u$). The p-(l, u)-partition problem can be decided in time $\mathcal{O}(u^2 p^2 n^2)$.

Results

RUB

Theorem
Given a weighted cactus graph G, a positive integer p and two non-negative integers l and u (with $l \leq u$). The p-(l,u)-partition problem can be decided in time ~~$\mathcal{O}(u^2 p^2 n^2)$~~. $\mathcal{O}(p^4 n^2)$

reduce size of partition sets from $\mathcal{O}(up)$ to $\mathcal{O}(p^2)$

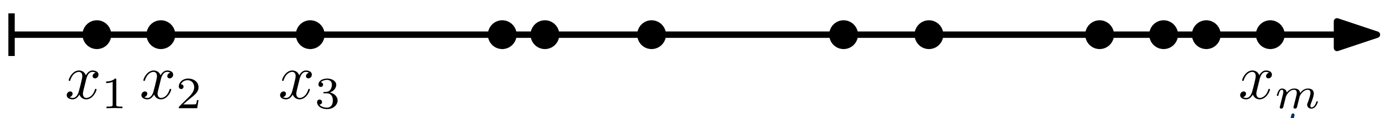
RUB

Theorem

Given a weighted cactus graph G, a positive integer p and two non-negative integers l and u (with $l \leq u$). The p-(l, u)-partition problem can be decided in time ~~$\mathcal{O}(u^2p^2n^2)$~~. $\mathcal{O}(p^4n^2)$

reduce size of partition sets from $\mathcal{O}(up)$ to $\mathcal{O}(p^2)$

before:



for each k $(\leq p)$ $\Rightarrow$ overall $\mathcal{O}(up)$ ↳ $m \leq u$

Results

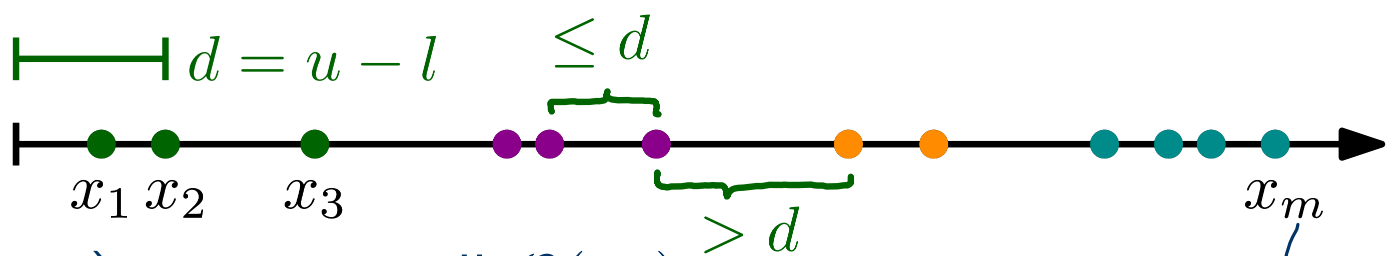
RUB

Theorem

Given a weighted cactus graph G, a positive integer p and two non-negative integers l and u (with $l \leq u$). The p-(l, u)-partition problem can be decided in time ~~$\mathcal{O}(u^2 p^2 n^2)$~~. $\mathcal{O}(p^4 n^2)$

reduce size of partition sets from $\mathcal{O}(up)$ to $\mathcal{O}(p^2)$

before:



for each k $(\leq p)$ $\Rrightarrow$ overall $\mathcal{O}(up)$ $\hookrightarrow m \leq u$

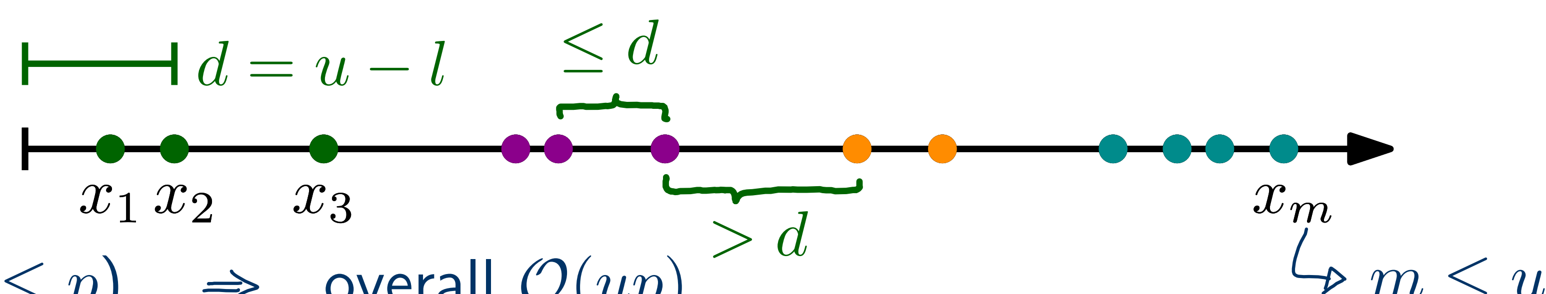
RUB

Theorem

Given a weighted cactus graph G, a positive integer p and two non-negative integers l and u (with $l \leq u$). The p-(l,u)-partition problem can be decided in time ~~$\mathcal{O}(u^2p^2n^2)$~~. $\mathcal{O}(p^4n^2)$

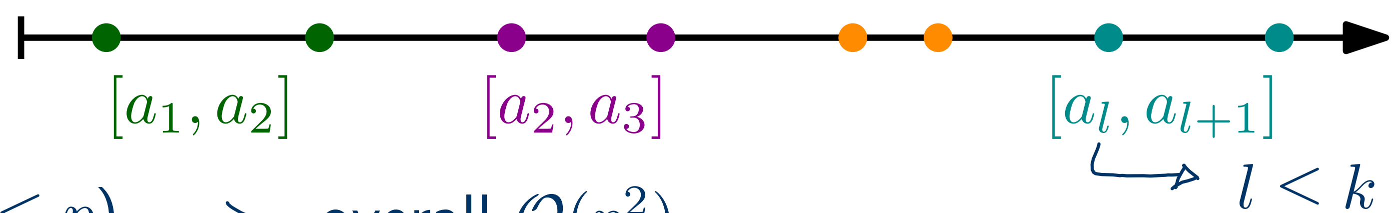
reduce size of partition sets from $\mathcal{O}(up)$ to $\mathcal{O}(p^2)$

before:



for each k $(\leq p)$ $\Rrightarrow$ overall $\mathcal{O}(up)$

now:



for each k $(\leq p)$ $\Rrightarrow$ overall $\mathcal{O}(p^2)$

Results

RUB

Theorem

Given a weighted cactus graph G, a positive integer p and two non-negative integers l and u (with $l \leq u$).
The p-(l, u)-partition problem can be ~~decided~~ solved in time $\mathcal{O}(p^4 n^2)$ and space $\mathcal{O}(p^5 n^2)$.

by storing additional information
and using backtracking

Results

RUB

Theorem

Given a weighted cactus graph G and two non-negative integers l and u (with $l \leq u$).
The minimum and maximum (l, u)-partition problem can solved in time $\mathcal{O}(n^6)$ and space $\mathcal{O}(n^7)$.

Results

Open problems

- NP-hard
- Polynomial-time algorithms

for other graph classes?