Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Maximum Betweenness Centrality: Approximability and Tractable Cases

Martin Fink and Joachim Spoerhase

Universität Würzburg

A Centrality Problem

Imagine an abstract network.

- computer network
- transportation network

This network can be modeled by a graph.

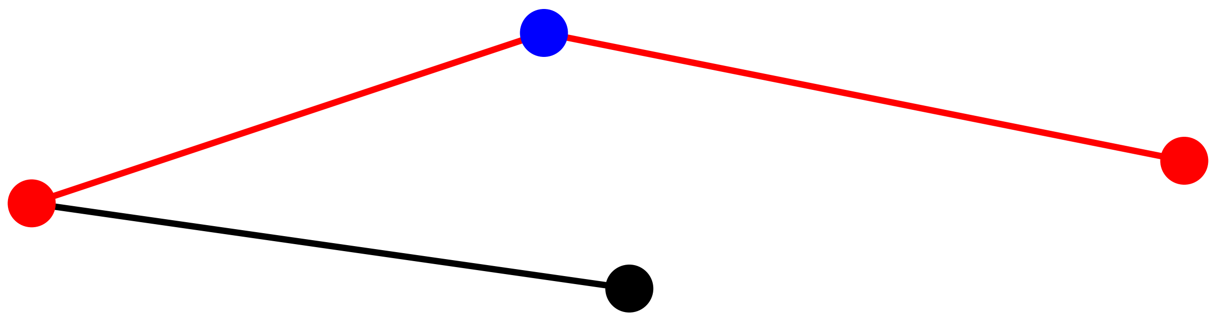
A Centrality Problem

Imagine an abstract network.

- computer network
- transportation network

This network can be modeled by a graph.

- Occupy some of the nodes.
- As much communication as possible should be detected.

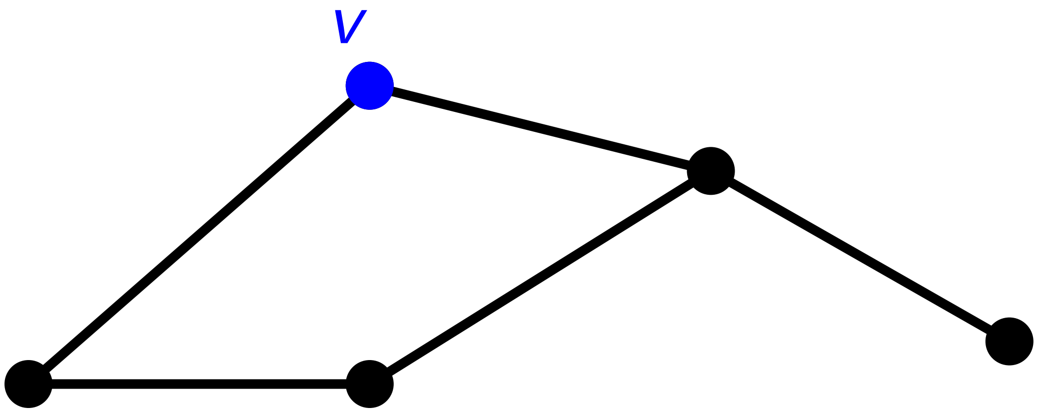


Overview

- Maximum Betweenness Centrality
- Approximating MBC
- APX-Completeness
- MBC on Trees
- Conclusion

Betweenness Centrality

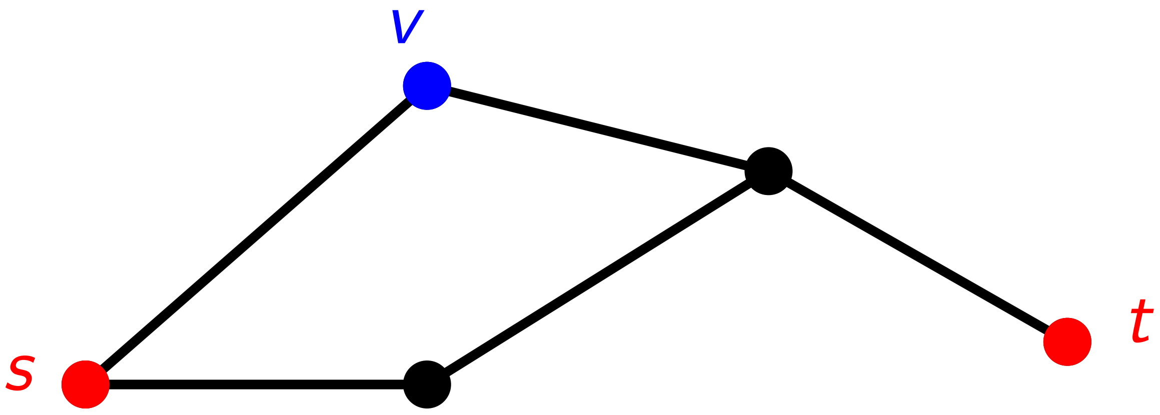
Given a graph $G = (V, E)$ and a node $v \in V$



Betweenness Centrality

Given a graph $G = (V, E)$ and a node $v \in V$

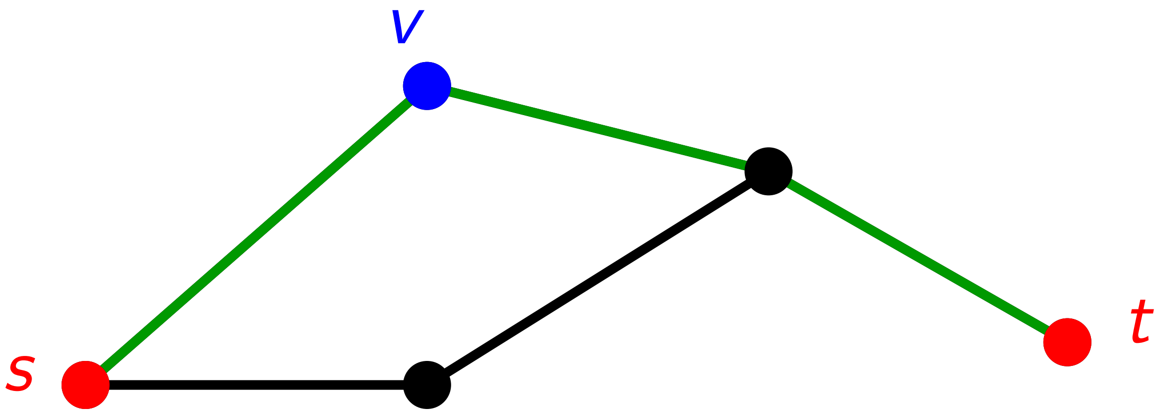
- choose communicating pair $s, t \in V$ uniformly at random



Betweenness Centrality

Given a graph $G = (V, E)$ and a node $v \in V$

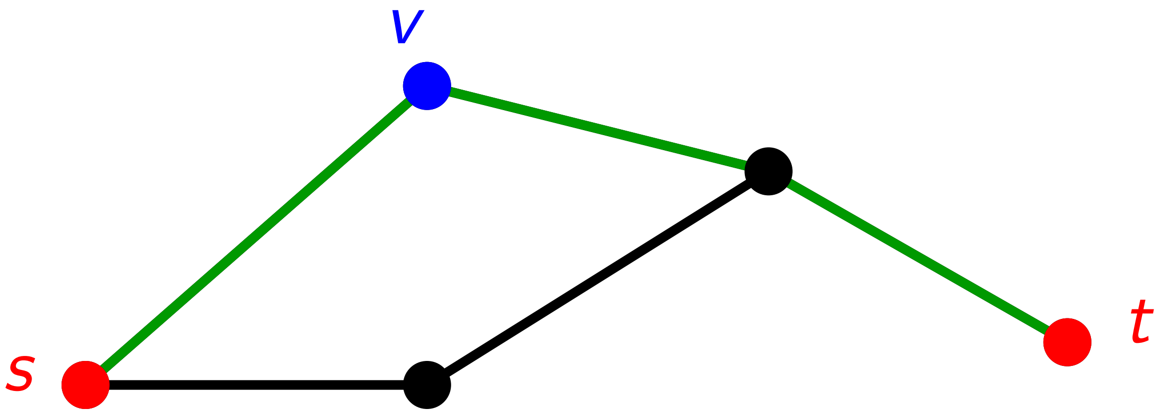
- choose communicating pair $s, t \in V$ uniformly at random
- choose one *shortest* s–t path P uniformly at random



Betweenness Centrality

Given a graph $G = (V, E)$ and a node $v \in V$

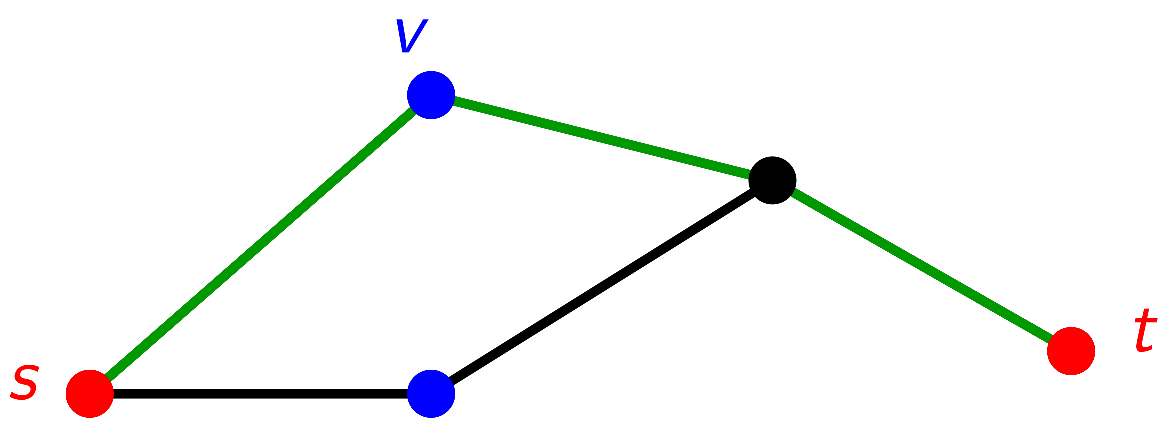
- choose communicating pair $s, t \in V$ uniformly at random
- choose one *shortest* s–t path P uniformly at random
- probability that v lies on P?



Group Betweenness Centrality

Given a graph $G = (V, E)$ and a ~~node $v \in V$~~ set $C \subseteq V$

- choose communicating pair $s, t \in V$ uniformly at random
- choose one *shortest* s–t path P uniformly at random
- probability that ~~v lies on P?~~ a node $v \in C$ lies on P?

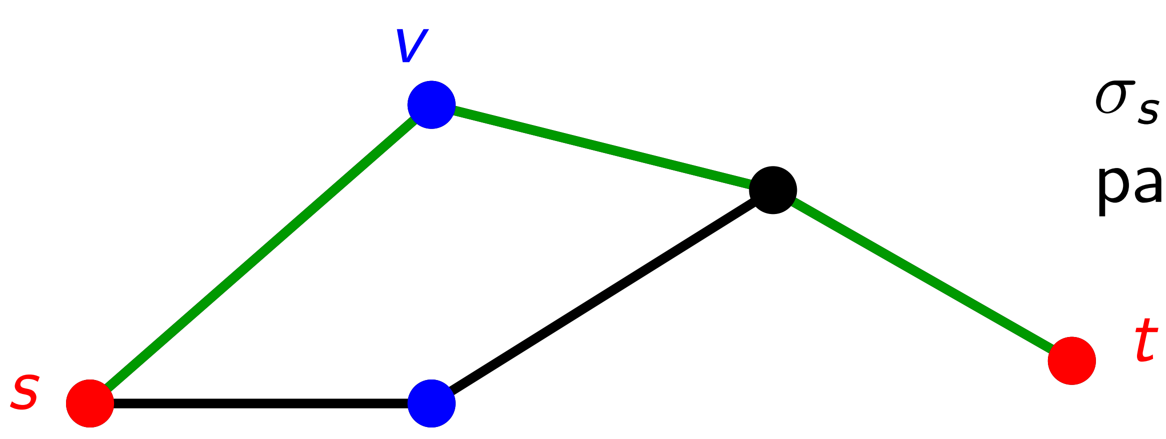


Group Betweenness Centrality

Given a graph $G = (V, E)$ and a ~~node $v \in V$~~ set $C \subseteq V$

- choose communicating pair $s, t \in V$ uniformly at random
- choose one *shortest* s–t path P uniformly at random
- probability that ~~v lies on P?~~ a node $v \in C$ lies on P?

$$\frac{\sigma_{s,t}(C)}{\sigma_{s,t}}$$



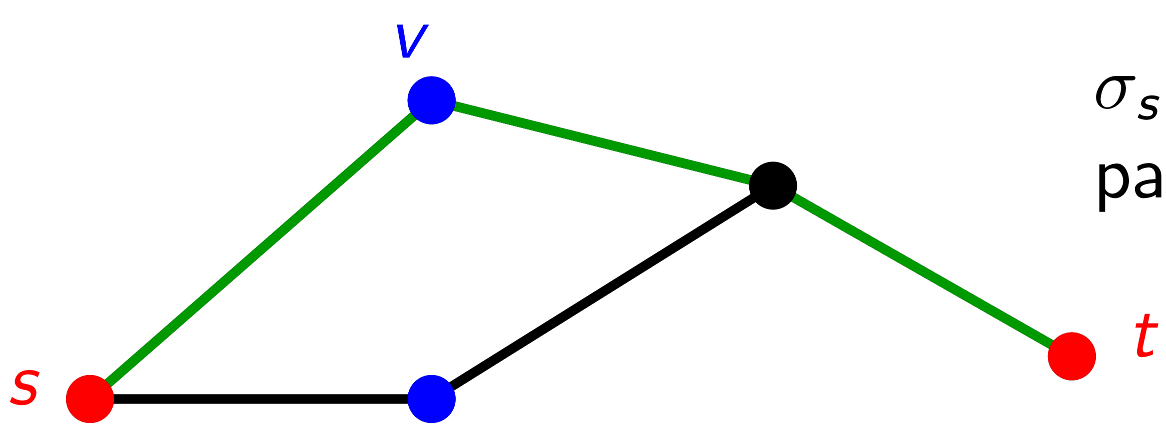
$\sigma_{s,t}$, $\sigma_{s,t}(C)$: #shortest s–t paths (using a node of C)

Group Betweenness Centrality

Given a graph $G = (V, E)$ and a ~~node $v \in V$~~ set $C \subseteq V$

- choose communicating pair $s, t \in V$ uniformly at random
- choose one *shortest* s–t path P uniformly at random
- probability that ~~v lies on P?~~ a node $v \in C$ lies on P?

$$\mathrm{GBC}(C) := \sum_{s,t \in V \mid s \neq t} \frac{\sigma_{s,t}(C)}{\sigma_{s,t}}$$



$\sigma_{s,t}$, $\sigma_{s,t}(C)$: #shortest s–t paths (using a node of C)

Previous Results

Theorem. [Brandes, 2001]
The Shortest Path Betweenness Centrality of all nodes can be computed in $O(nm)$ time.

Theorem. [Puzis et. al., 2007]
The Group Betweenness Centrality of one set $C \subseteq V$ can be computed in $O(n^3)$ time.

Previous Results

Theorem. [Brandes, 2001]
The Shortest Path Betweenness Centrality of all nodes can be computed in $O(nm)$ time.

Theorem. [Puzis et. al., 2007]
The Group Betweenness Centrality of one set $C \subseteq V$ can be computed in $O(n^3)$ time.

Method: iteratively add nodes, $O(n^2)$ update time for each step

Maximum Betweenness Centrality

Input: A Graph $G = (V, E)$, node costs $c : V \to \mathbb{R}_0^+$, budget $b \in \mathbb{R}_0^+$

Maximum Betweenness Centrality

Input: A Graph $G = (V, E)$, node costs $c : V \to \mathbb{R}_0^+$, budget $b \in \mathbb{R}_0^+$

Task: Find a set $C \subseteq V$ with $c(C) \leq b$ maximizing $\mathrm{GBC}(C)$

Maximum Betweenness Centrality

Input: A Graph $G = (V, E)$, node costs $c : V \to \mathbb{R}_0^+$, budget $b \in \mathbb{R}_0^+$

Task: Find a set $C \subseteq V$ with $c(C) \leq b$ maximizing GBC(C)

Theorem. [Puzis et al., 2007]
(unit-cost) MBC is NP-hard.

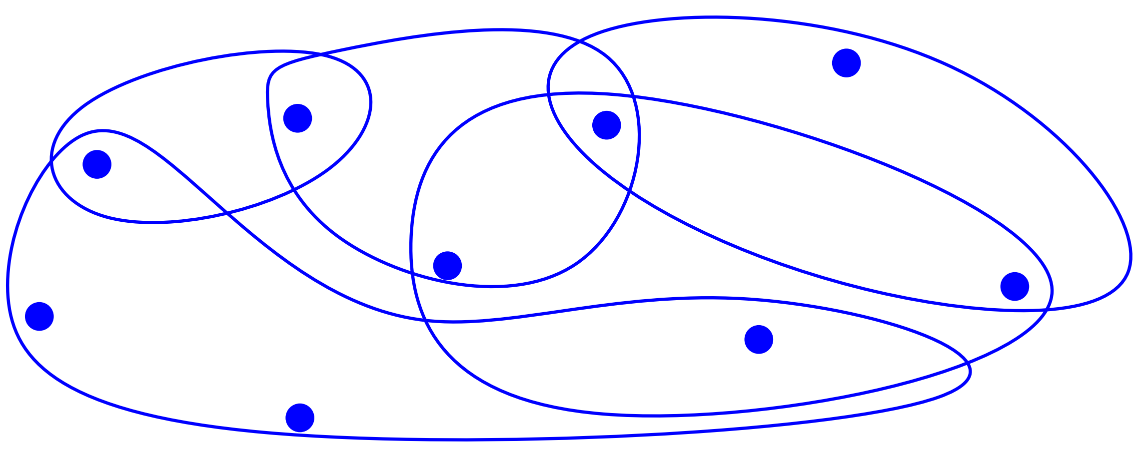
Theorem. [Dolev et al., 2009]
A simple greedy-algorithm computes a $(1 - 1/e)$-approximation for *unit-cost* MBC in $O(n^3)$ time.

Approximating MBC

- Reduce MBC to (budgeted) Maximum Coverage.
- Use existing results for Maximum Coverage.
- implicit reduction

(budgeted) Maximum Coverage

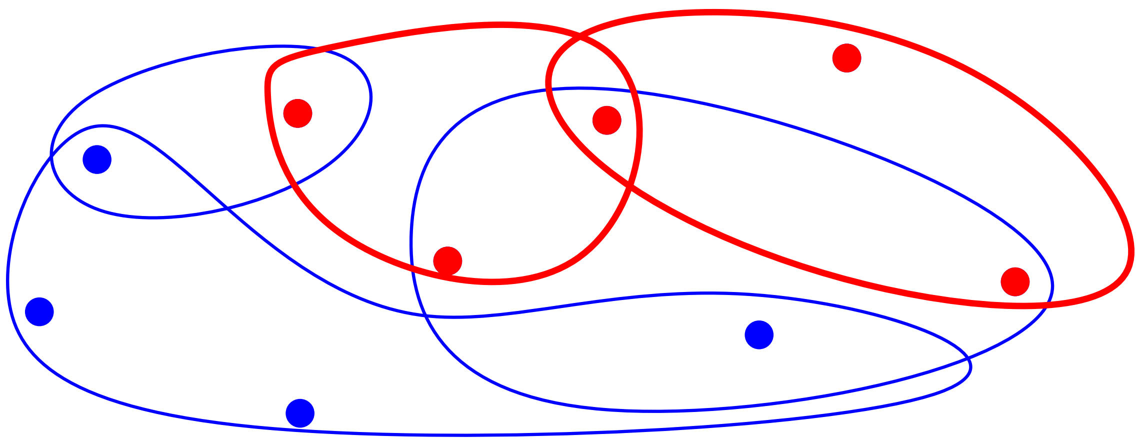
Input: set S, weight function $w\colon S \to \mathbb{R}_0^+$
family $\mathcal{F}$ of subsets of S;
costs $c'\colon \mathcal{F} \to \mathbb{R}_0^+$ and a budget $b \geq 0$



(budgeted) Maximum Coverage

Input: set S, weight function $w \colon S \to \mathbb{R}_0^+$
family $\mathcal{F}$ of subsets of S;
costs $c' \colon \mathcal{F} \to \mathbb{R}_0^+$ and a budget $b \geq 0$

Task: Find a collection $C' \subseteq \mathcal{F}$ with $c'(C') \leq b$ maximizing the total weight $w(C')$ of the ground elements covered by C'



(budgeted) Maximum Coverage and MBC

Input: set S, weight function $w\colon S \to \mathbb{R}_0^+$
family $\mathcal{F}$ of subsets of S;
costs $c'\colon \mathcal{F} \to \mathbb{R}_0^+$ and a budget $b \geq 0$

Task: Find a collection $C' \subseteq \mathcal{F}$ with $c'(C') \leq b$ maximizing the total weight $w(C')$ of the ground elements covered by C'

shortest s–t path P

weight $w(P) := \frac{1}{\sigma_{s,t}}$

(budgeted) Maximum Coverage and MBC

Input: set S, weight function $w\colon S \to \mathbb{R}_0^+$
family $\mathcal{F}$ of subsets of S;
costs $c'\colon \mathcal{F} \to \mathbb{R}_0^+$ and a budget $b \geq 0$

Task: Find a collection $C' \subseteq \mathcal{F}$ with $c'(C') \leq b$ maximizing the total weight $w(C')$ of the ground elements covered by C'

$v \in V$: set $S(v)$ of all shortest paths containing v

costs $c'(S(v)) = c(v)$

shortest s–t path P

weight $w(P) := \frac{1}{\sigma_{s,t}}$

(budgeted) Maximum Coverage and MBC

Input: set S, weight function $w\colon S \to \mathbb{R}_0^+$
family $\mathcal{F}$ of subsets of S;
costs $c'\colon \mathcal{F} \to \mathbb{R}_0^+$ and a budget $b \geq 0$

Task: Find a collection $C' \subseteq \mathcal{F}$ with $c'(C') \leq b$ maximizing the total weight $w(C')$ of the ground elements covered by C'

$v \in V$: set $S(v)$ of all shortest paths containing v

costs $c'(S(v)) = c(v)$

for set $C \subseteq V$: $w(S(C)) = \mathrm{GBC}(C)$

shortest s–t path P

weight $w(P) := \frac{1}{\sigma_{s,t}}$

Approximation Algorithms for MBC

$U := V$

while $U \neq \emptyset$ **do**

$\quad u =$ node with maximal $\frac{\text{GBC}(C+u)-\text{GBC}(C)}{c(u)}$

$\quad$ **if** $c(C+u) \leq b$ **then**

$\quad\quad C := C + u$

$\quad U := U - u$

Approximation Algorithms for MBC

Theorem. [Dolev et al., 2009]
$(1-1/e)$-approximation for *unit-cost* MBC in $O(n^3)$ time.

$U := V$
while $U \neq \emptyset$ **do**
  u = node with maximal $\frac{\mathrm{GBC}(C+u)-\mathrm{GBC}(C)}{c(u)}$
  if $c(C+u) \leq b$ **then**
    $C := C + u$
  $U := U - u$

Approximation Algorithms for MBC

Theorem. [Dolev et al., 2009]
$(1-1/e)$-approximation for *unit-cost* MBC in $O(n^3)$ time.

reduction to Maximum Coverage simplifies the proof

$U := V$
while $U \neq \emptyset$ **do**
  $u =$ node with maximal $\frac{\mathrm{GBC}(C+u)-\mathrm{GBC}(C)}{c(u)}$
  if $c(C+u) \leq b$ **then**
    $C := C + u$
  $U := U - u$

Theorem. [Khuller et al., 1999]
simple greedy approach: $(1-1/\sqrt{e})$-approximation for Maximum Coverage ($(1-1/e)$ for unit-cost version)

Approximation Algorithms for MBC

Theorem. [Dolev et al., 2009]
$(1 - 1/e)$-approximation for *unit-cost* MBC in $O(n^3)$ time.

reduction to Maximum Coverage simplifies the proof

$U := V$
while $U \neq \emptyset$ **do**
  $u =$ node with maximal $\frac{\mathrm{GBC}(C+u) - \mathrm{GBC}(C)}{c(u)}$
  if $c(C + u) \leq b$ **then**
    $C := C + u$
  $U := U - u$

better approximation for arbitrary costs?

Theorem. [Khuller et al., 1999]
simple greedy approach: $(1 - 1/\sqrt{e})$-approximation for Maximum Coverage ($(1 - 1/e)$ for unit-cost version)

Approximation Algorithms for MBC

Extended greedy approach

```
H := ∅
foreach C ⊆ V with |C| ≤ 3 and c(C) ≤ b do
    U := V \ C
    while U ≠ ∅ do
        u = node with maximal (GBC(C+u) − GBC(C)) / c(u)
        if c(C + u) ≤ b then
            C := C + u
        U := U − u
    if GBC(C) > GBC(H) then H := C
return H
```

Approximation Algorithms for MBC

Theorem. [Khuller et al., 1999]
The extended greedy approach yields an approximation factor of $(1 - 1/e)$ for Maximum Coverage.

Approximation Algorithms for MBC

Theorem. [Khuller et al., 1999]
The extended greedy approach yields an approximation factor of $(1 - 1/e)$ for Maximum Coverage.

reduction

Theorem. A $(1 - 1/e)$-approximative solution for MBC can be computed in $O(n^6)$ using the extended greedy approach.

Approximation Algorithms for MBC

Theorem. [Khuller et al., 1999]
The extended greedy approach yields an approximation factor of $(1 - 1/e)$ for Maximum Coverage.

reduction

Theorem. A $(1 - 1/e)$-approximative solution for MBC can be computed in $O(n^6)$ using the extended greedy approach.

Theorem. [Khuller et al., 1999]
The approximation factor of $(1 - 1/e)$ of the greedy algorithm for Maximum Coverage is tight.

Approximation Algorithms for MBC

Theorem. [Khuller et al., 1999]
The extended greedy approach yields an approximation factor of $(1 - 1/e)$ for Maximum Coverage.

reduction

Theorem. A $(1 - 1/e)$-approximative solution for MBC can be computed in $O(n^6)$ using the extended greedy approach.

Theorem. [Khuller et al., 1999]
The approximation factor of $(1 - 1/e)$ of the greedy algorithm for Maximum Coverage is tight.

Theorem. The approximation factor of $(1 - 1/e)$ of the greedy algorithm for MBC is tight.

MBC is APX-complete

Maximum Vertex Cover:

Input: Graph $G = (V, E)$, number $k \leq n = |V|$

Task: find a set $C \subseteq V$ with $|C| = k$ maximizing the number of covered edges

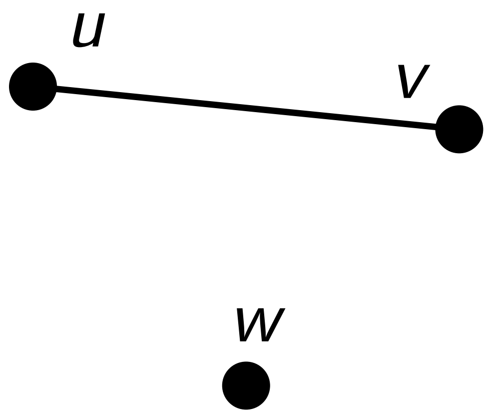
MBC is APX-complete

Maximum Vertex Cover:

Input: Graph $G = (V, E)$, number $k \leq n = |V|$

Task: find a set $C \subseteq V$ with $|C| = k$ maximizing the number of covered edges

Maximum Vertex Cover



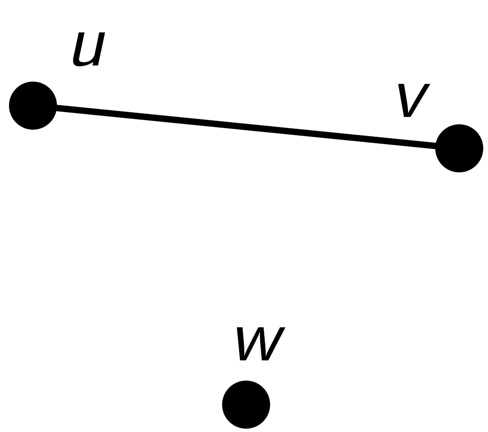
MBC is APX-complete

Maximum Vertex Cover:

Input: Graph $G = (V, E)$, number $k \leq n = |V|$

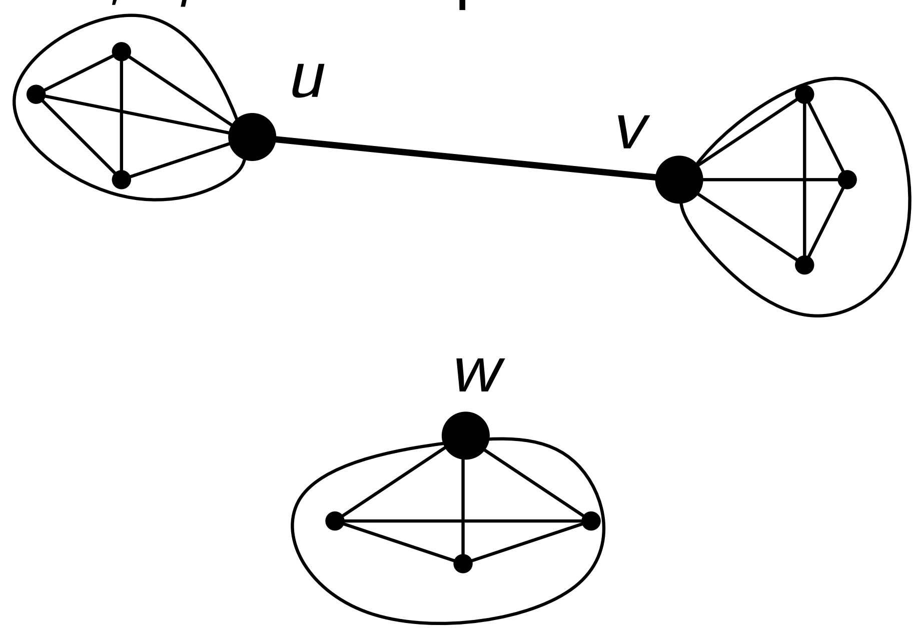
Task: find a set $C \subseteq V$ with $|C| = k$ maximizing the number of covered edges

Maximum Vertex Cover



MBC

copies $u_1, \ldots, u_l$ in a clique



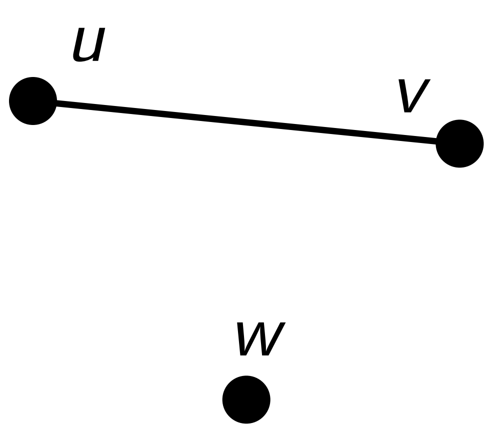
MBC is APX-complete

Maximum Vertex Cover:

Input: Graph $G = (V, E)$, number $k \leq n = |V|$

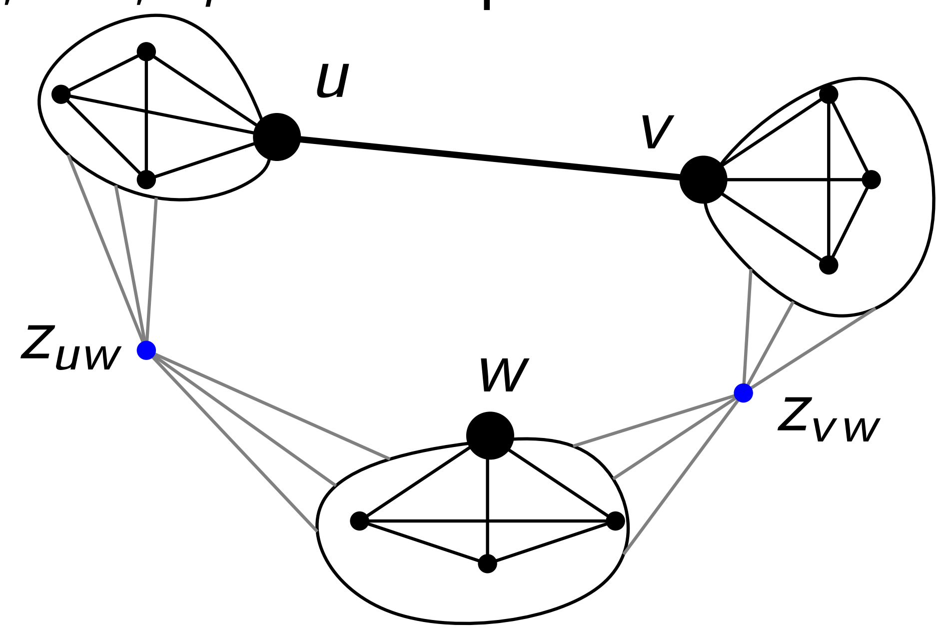
Task: find a set $C \subseteq V$ with $|C| = k$ maximizing the number of covered edges

Maximum Vertex Cover



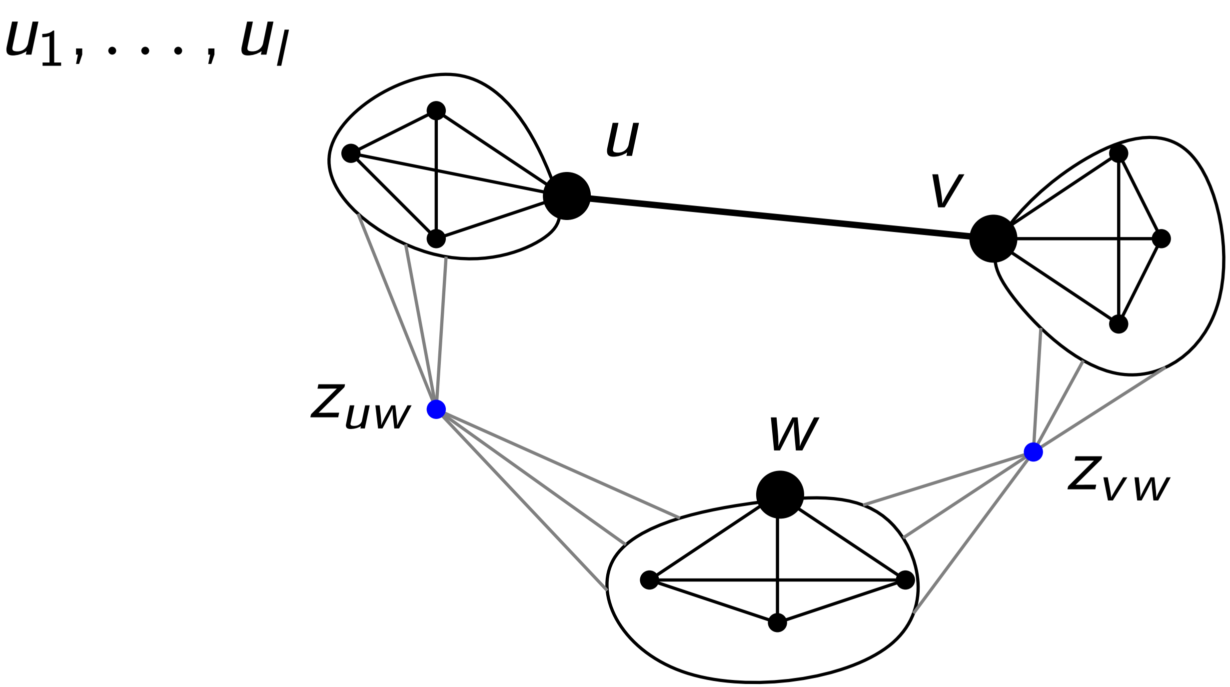
MBC

copies $u_1, \ldots, u_l$ in a clique



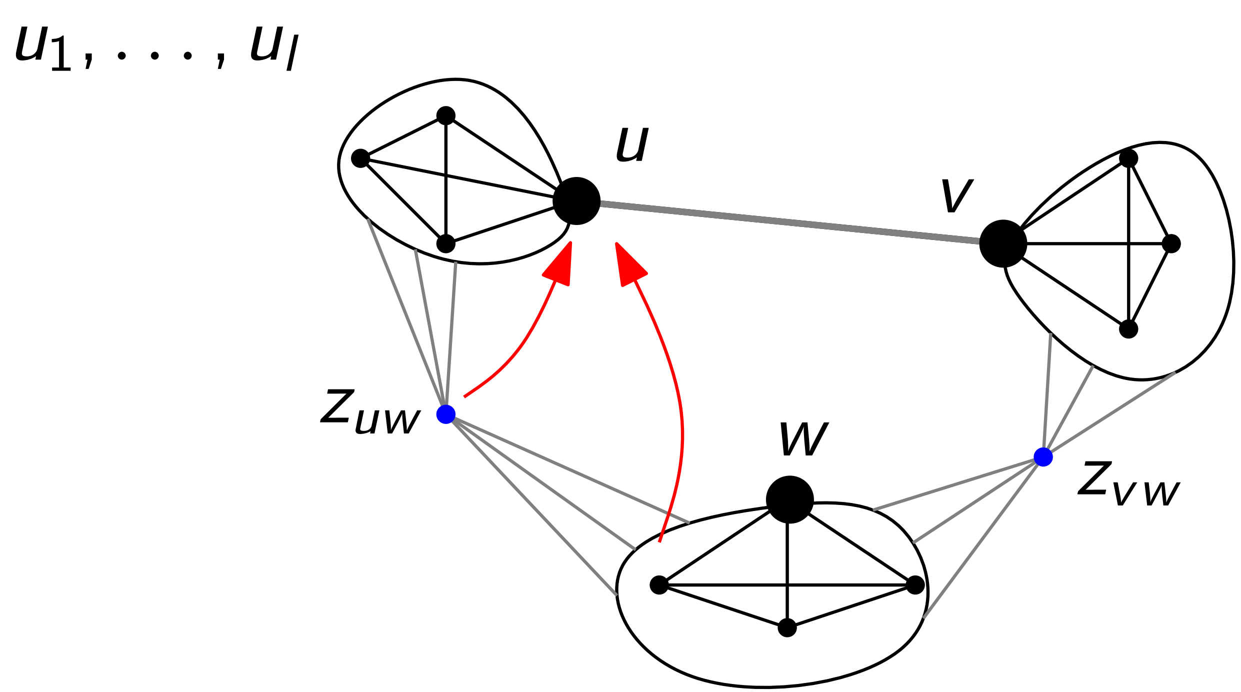
MBC is APX-complete

- Only paths between copies of distinct nodes are essential (for large l):
  - u covers shortest path for all l^2 pairs (u_i, v_j)
  - number of other pairs only linear in l



MBC is APX-complete

- Only paths between copies of distinct nodes are essential (for large l):
  - u covers shortest path for all l^2 pairs (u_i, v_j)
  - number of other pairs only linear in l
- Only original nodes from G are relevant candidates for the inclusion in set C with high GBC.

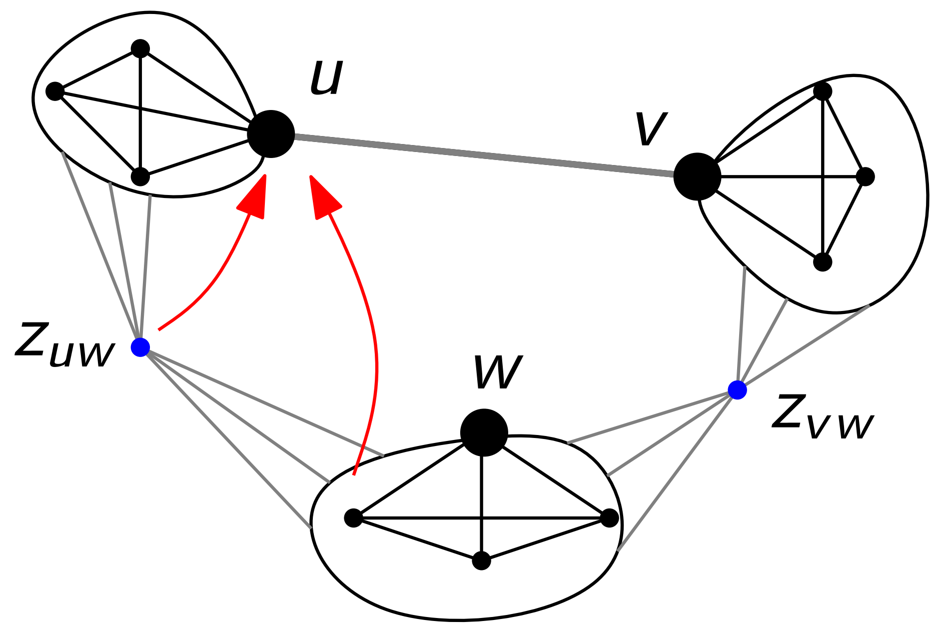


MBC is APX-complete

- Only paths between copies of distinct nodes are essential (for large l):
  - u covers shortest path for all l^2 pairs (u_i, v_j)
  - number of other pairs only linear in l
- Only original nodes from G are relevant candidates for the inclusion in set C with high GBC.

For $C \subseteq V$:

- $\mathrm{GBC}(C) \approx l^2 \times$ #covered edges in G

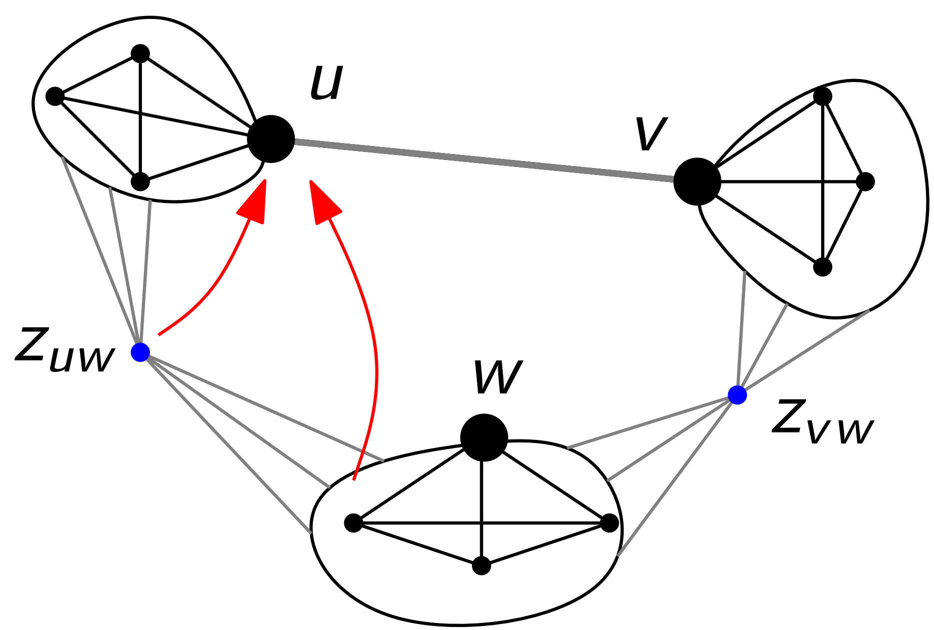


MBC is APX-complete

- Only paths between copies of distinct nodes are essential (for large l):
  - u covers shortest path for all l^2 pairs (u_i, v_j)
  - number of other pairs only linear in l
- Only original nodes from G are relevant candidates for the inclusion in set C with high GBC.

For $C \subseteq V$:

- $\mathrm{GBC}(C) \approx l^2 \times$ #covered edges in G
- C approximative solution for MBC $\Rightarrow$ C approximative solution for Maximum Vertex Cover



MBC is APX-complete

Theorem. [Petrank, 1994]
Maximum Vertex Cover is APX-complete.

Theorem. (Unit-cost) Maximum Betweenness Centrality is APX-complete.

Not much hope for a PTAS

MBC on Trees

- For tree $T = (V, E)$: Exactly one (shortest) path between each pair of nodes.

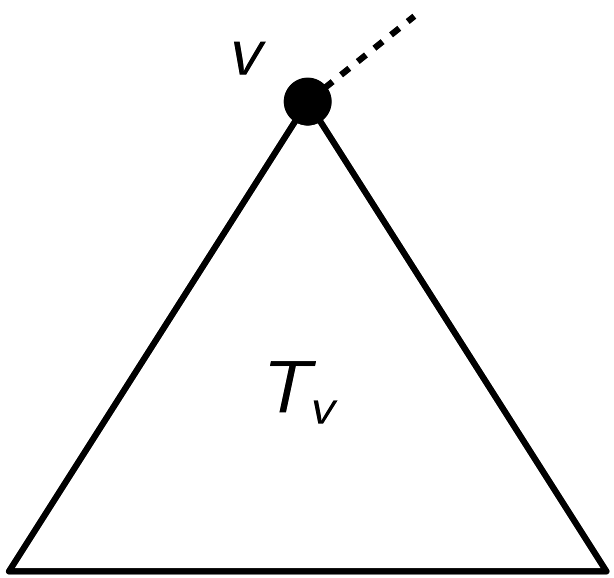
$$\text{GBC}(C) = \ \#\text{paths covered by } C$$

MBC on Trees

- For tree $T = (V, E)$: Exactly one (shortest) path between each pair of nodes.

$$\text{GBC}(C) = \#\text{paths covered by } C$$

- Use dynamic programming.



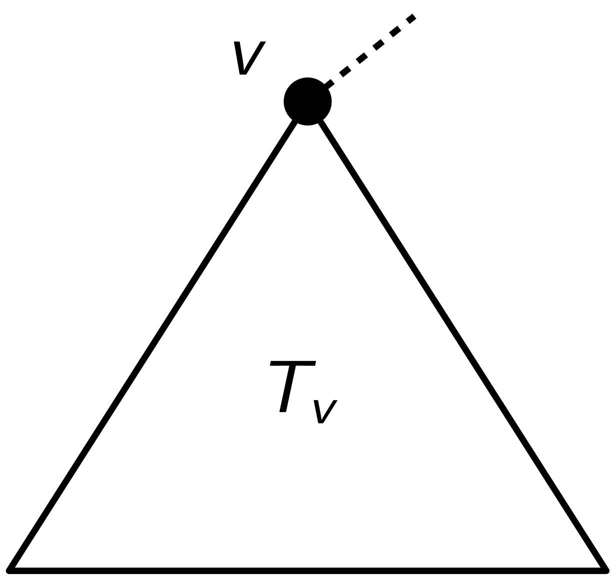
- $\text{GBC}_v(C) = \#\text{internal paths in } T_v$ covered by C

MBC on Trees

- For tree $T = (V, E)$: Exactly one (shortest) path between each pair of nodes.

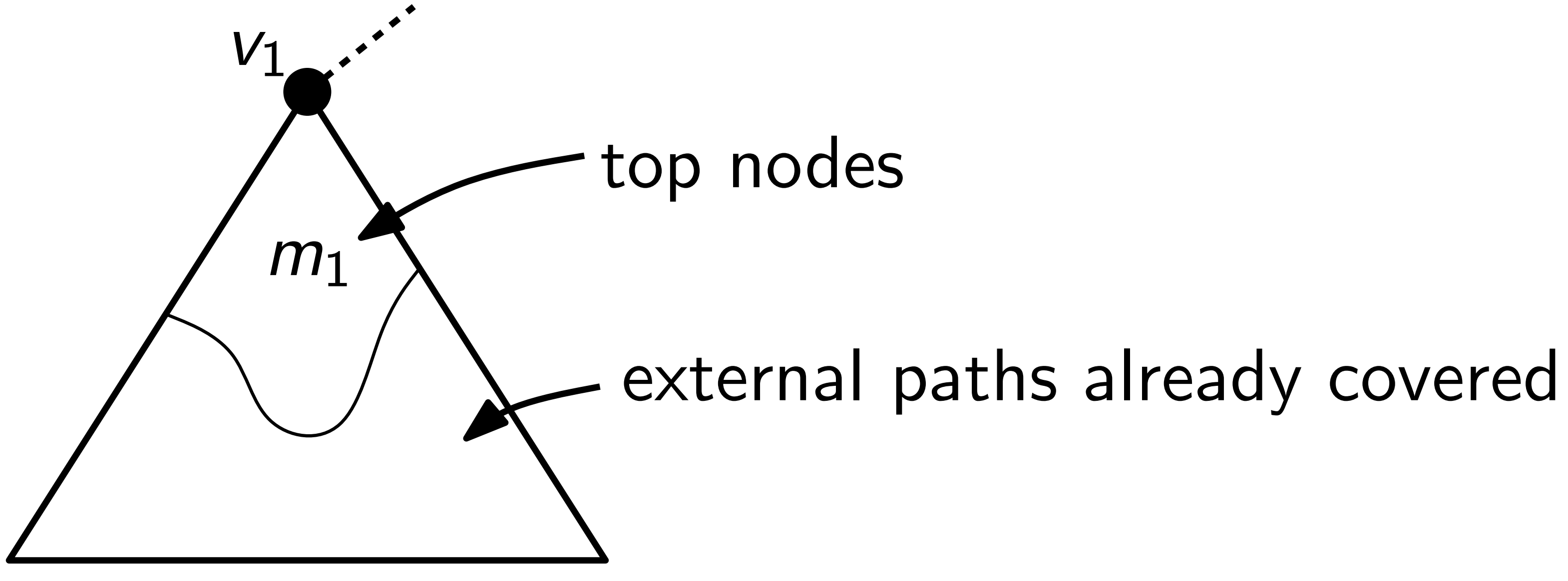
$$\mathrm{GBC}(C) = \text{ \#paths covered by } C$$

- Use dynamic programming.

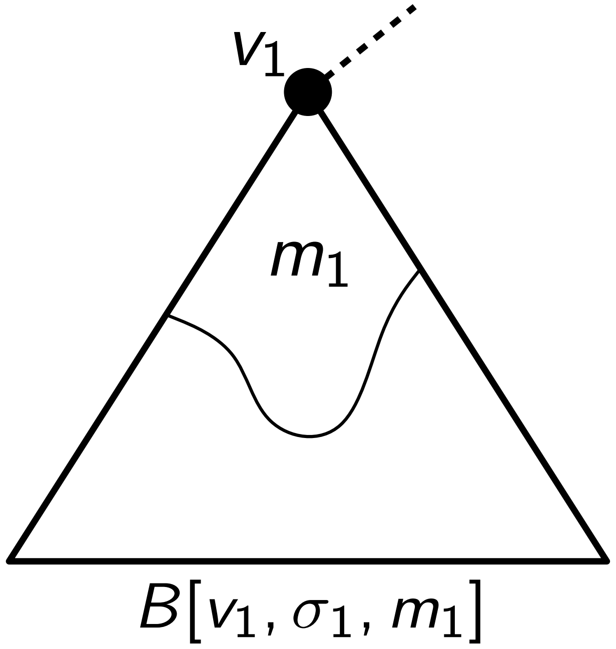


- $\mathrm{GBC}_v(C) = $ #internal paths in T_v covered by C
- Some paths from T_v to nodes outside might already be covered.

MBC on Trees

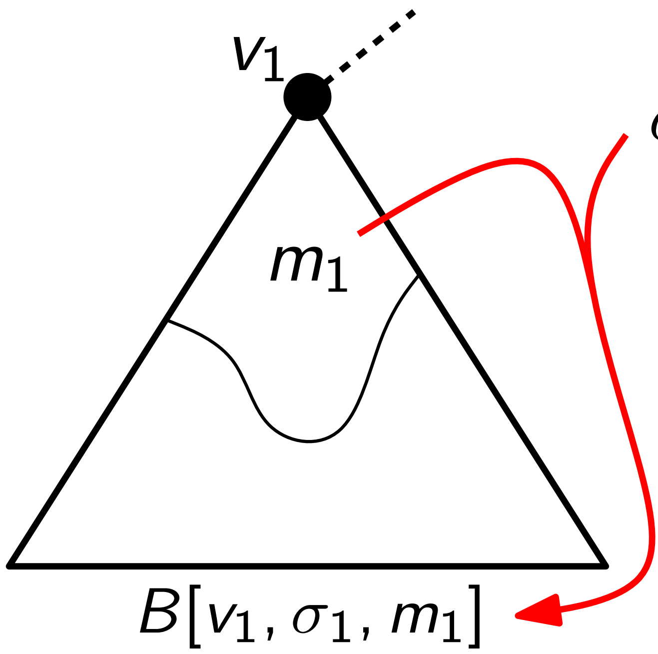


MBC on Trees



$\sigma_1 \leq \mathrm{GBC}_{v_1}(C) \leq n^2$ internal GBC

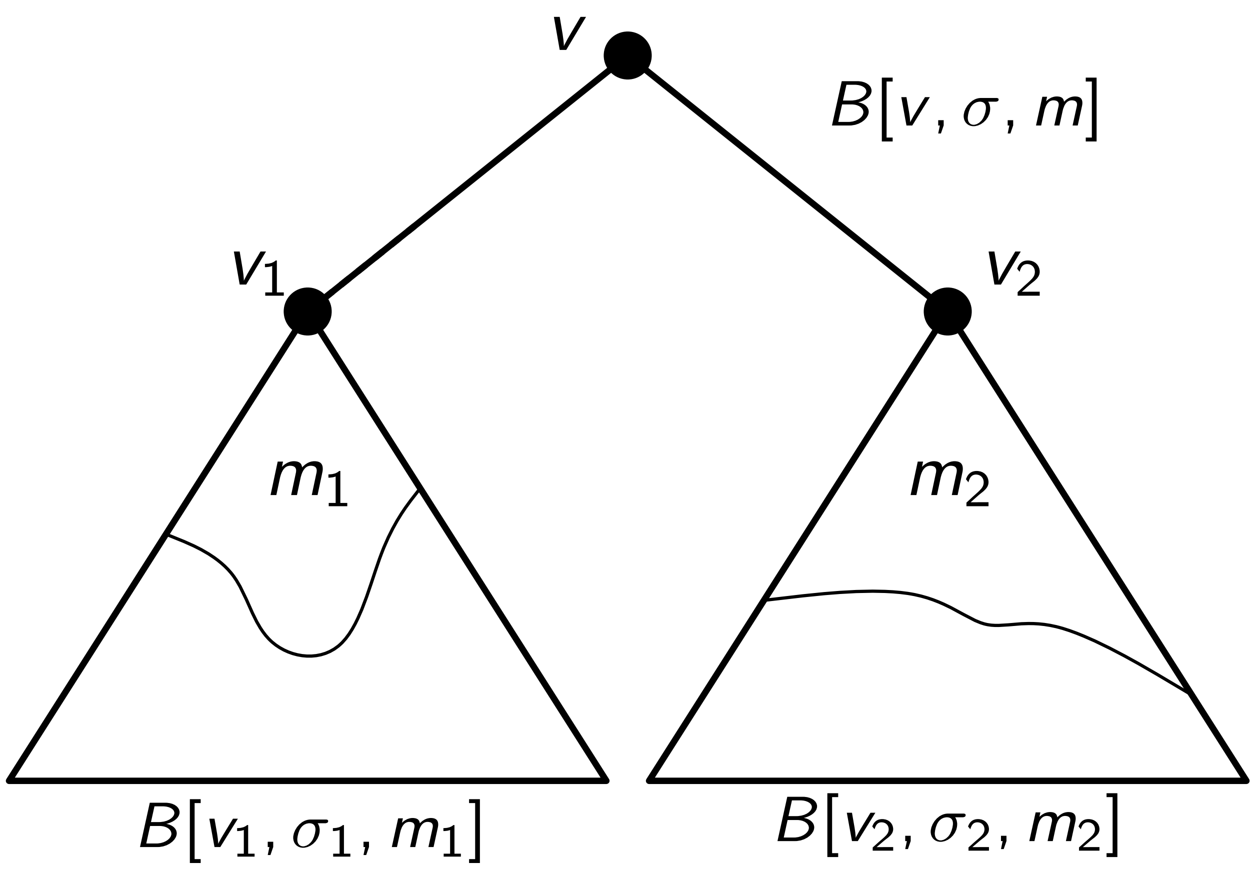
MBC on Trees



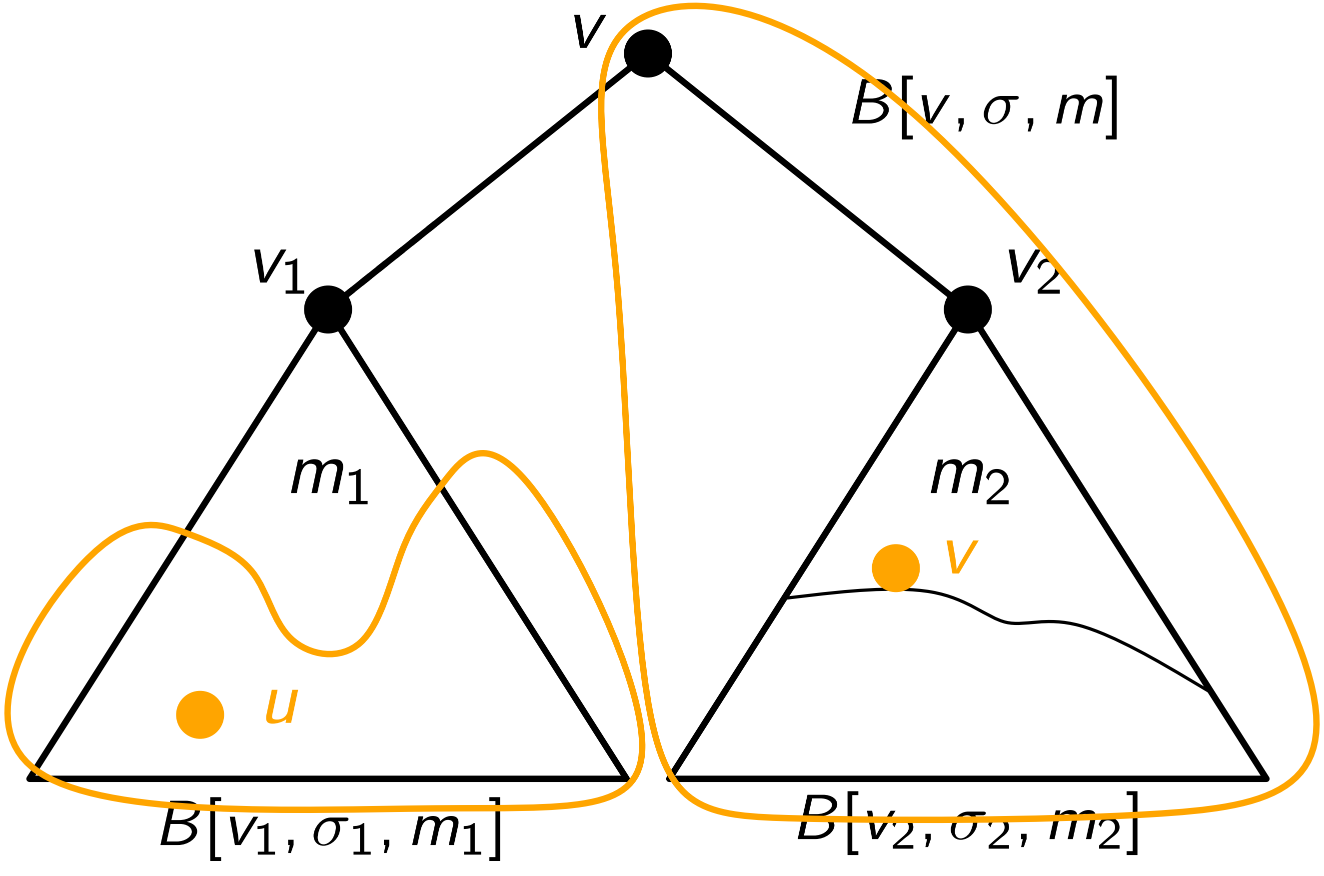
$\sigma_1 \leq \text{GBC}_{v_1}(C) \leq n^2$ internal GBC

cost of cheapest set $C \subseteq V$ providing these values σ_1, m_1

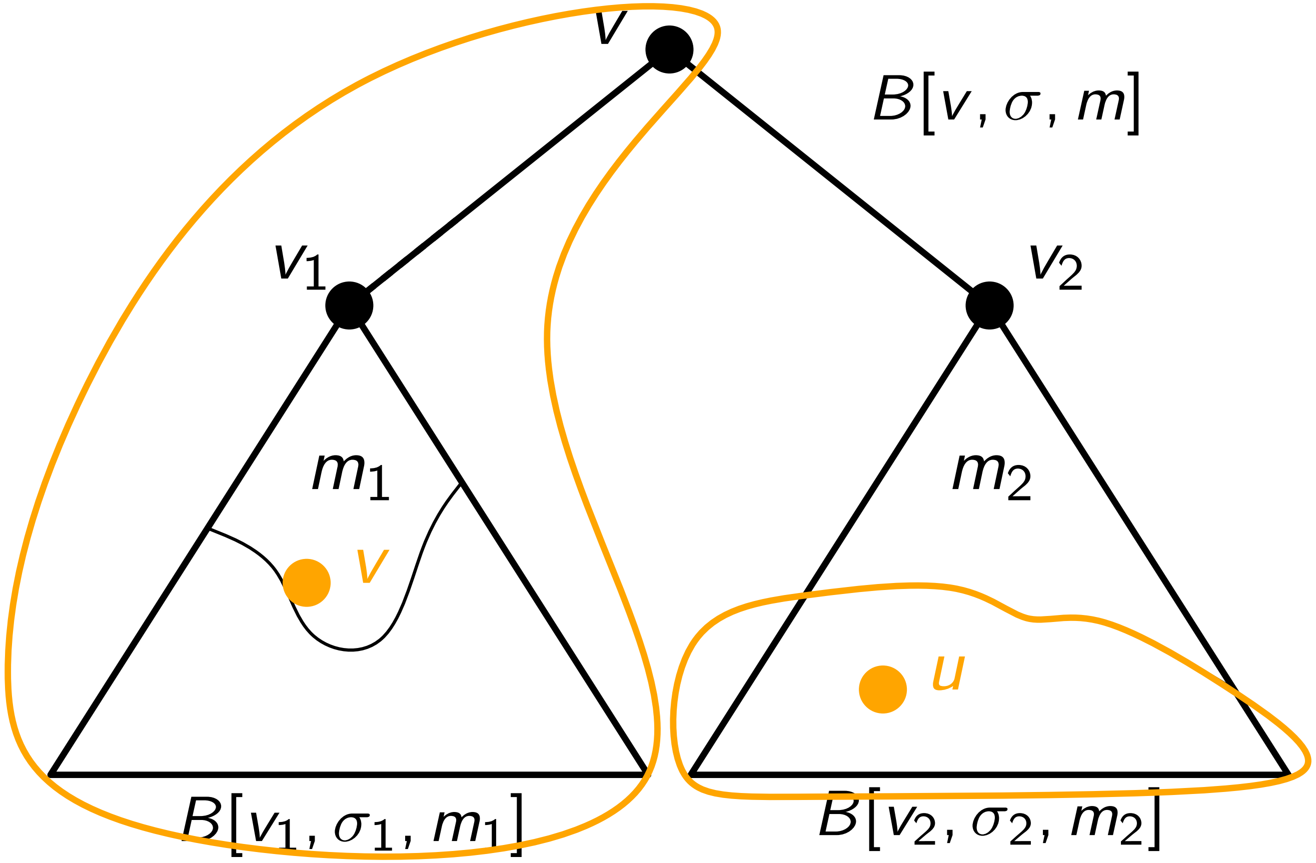
MBC on Trees



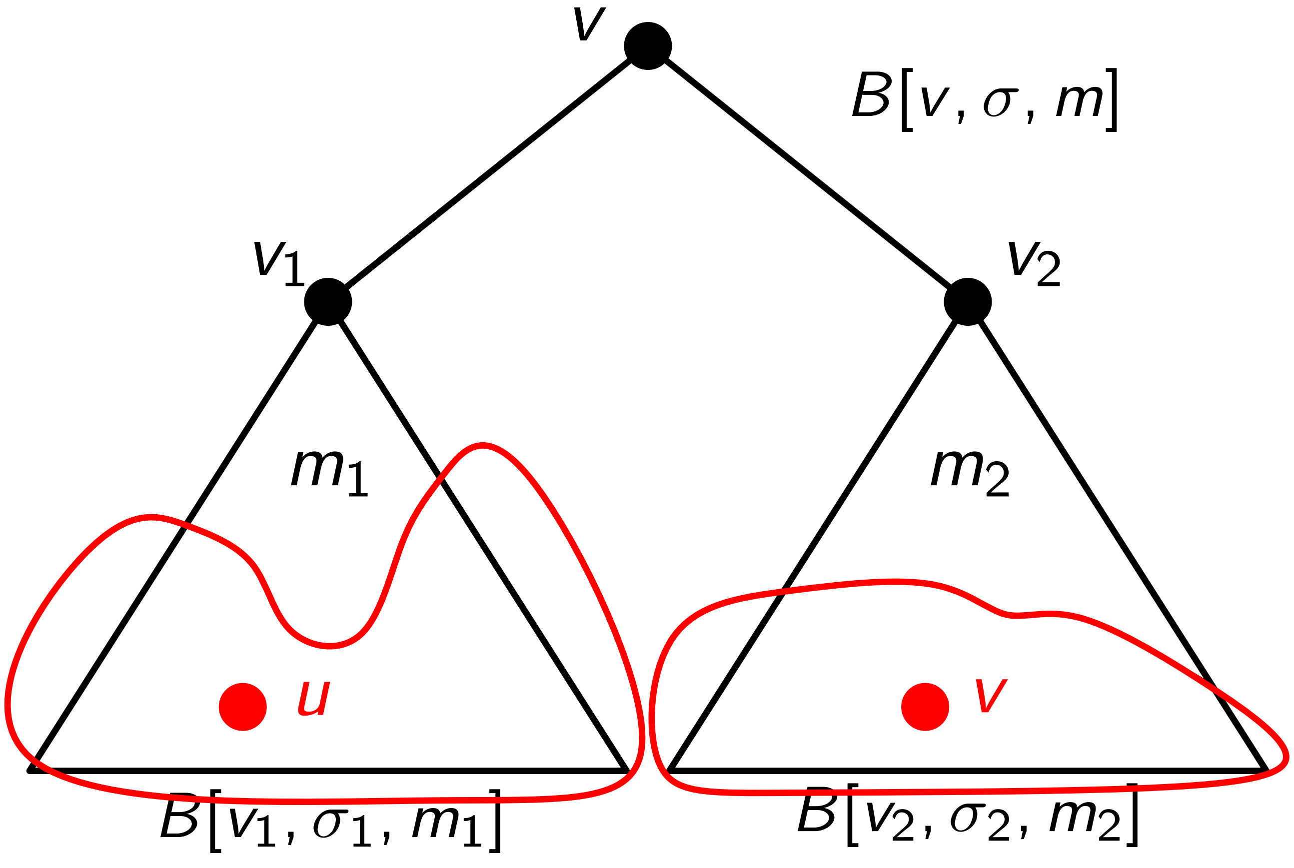
MBC on Trees



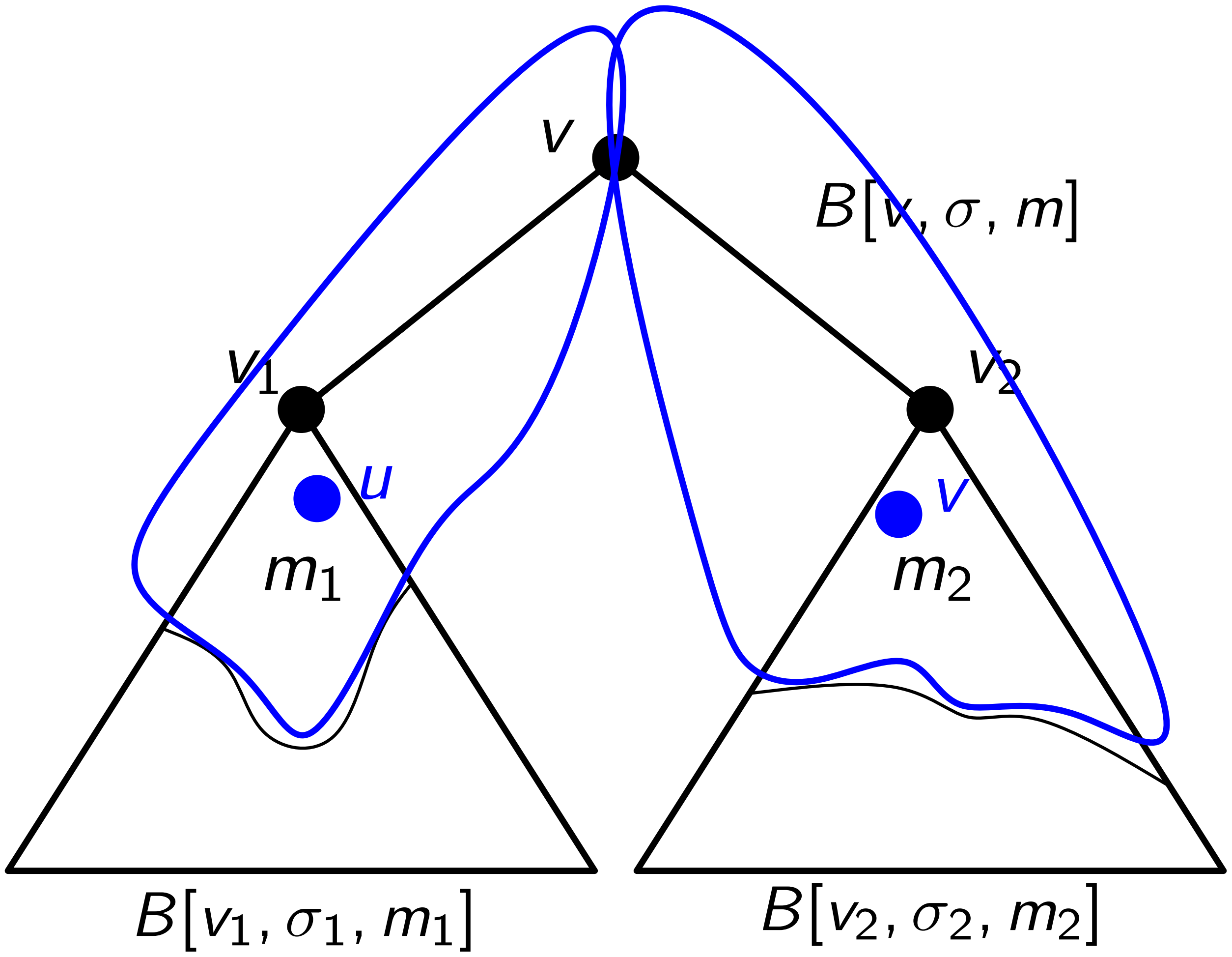
MBC on Trees



MBC on Trees



MBC on Trees

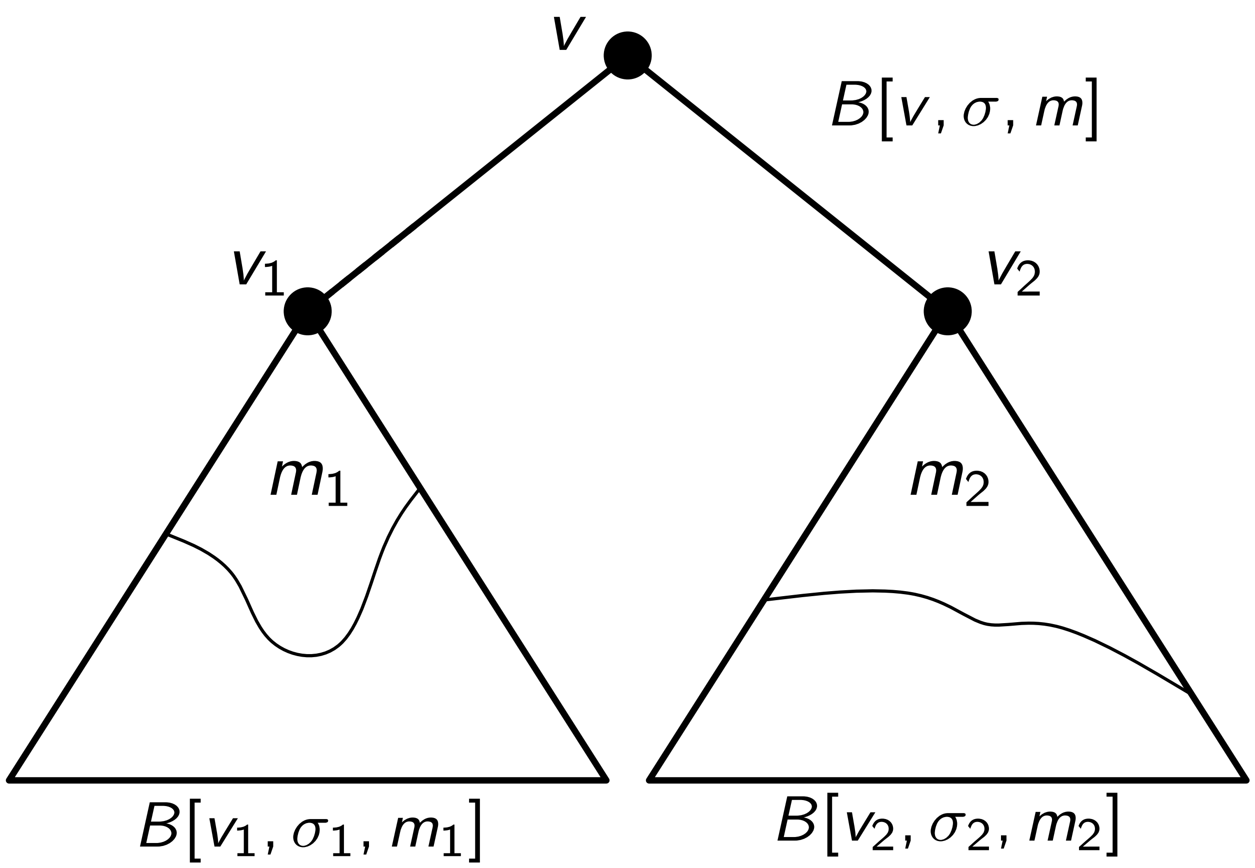


MBC on Trees

Computation of $B[v, \sigma, m]$:

- split m, σ among T_{v_1}, T_{v_2}, v
- $m = 0$ needs special handling

} $O(n^3)$ combinations



MBC on Trees

Computation of $B[v, \sigma, m]$:

- split m, σ among T_{v_1}, T_{v_2}, v
- $m = 0$ needs special handling

$\Big\}$ $O(n^3)$ combinations

(v, σ, m): $O(n \cdot n^2 \cdot n) = O(n^4)$ combinations

MBC on Trees

Computation of $B[v, \sigma, m]$:

- split m, σ among T_{v_1}, T_{v_2}, v
- $m = 0$ needs special handling

$\Big\}$ $O(n^3)$ combinations

(v, σ, m): $O(n \cdot n^2 \cdot n) = O(n^4)$ combinations

Theorem. MBC can be solved in $O(n^7)$ time on trees.

Conclusion

- Approximation Algorithm for Maximum Betweenness Centrality: tight approximation factor of $1 - 1/e$

Conclusion

- Approximation Algorithm for Maximum Betweenness Centrality: tight approximation factor of $1 - 1/e$

- Approximability

$$1 - \frac{1}{e} < \qquad \xcancel{1 - \epsilon}$$

Conclusion

- Approximation Algorithm for Maximum Betweenness Centrality: tight approximation factor of $1 - 1/e$

- Approximability

$$1 - \frac{1}{e} < \overset{?}{\alpha} < \cancel{1 - \epsilon}$$

not possible for Maximum Coverage

Conclusion

- Approximation Algorithm for Maximum Betweenness Centrality: tight approximation factor of $1 - 1/e$

- Approximability

$$1 - \frac{1}{e} < \overset{?}{\alpha} < \cancel{1 - \epsilon}$$

not possible for Maximum Coverage

- Polynomial-time Algorithm for trees

  Also possible for other classes of graphs?

Conclusion

- Approximation Algorithm for Maximum Betweenness Centrality: tight approximation factor of $1 - 1/e$
- Approximability

$$1 - \frac{1}{e} < \overset{?}{\alpha} < \cancel{1 - \epsilon}$$

not possible for Maximum Coverage

- Polynomial-time Algorithm for trees

  Also possible for other classes of graphs?

Thank you!