Weighted Upper Edge Cover: complexity and approximability

Mehdi Khosravian

Joint work with: Kaveh Khoshkhah, Jérôme Monnot and Florian Sikora

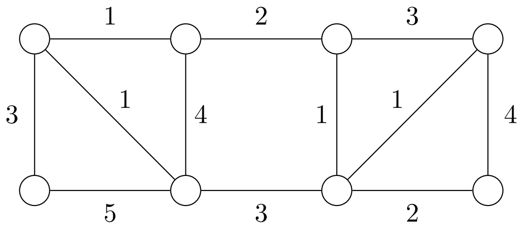
March 2, 2019

Min weighted Edge Cover Problem

Input: A Graph $G = (V, E, w)$ with $w(e) \geq 0$ for all $e \in E$.

Solution: A subset $S \subseteq E$ which covers all vertices in V.

Output: Minimizing $w(S) = \sum_{e \in S} w(e)$.

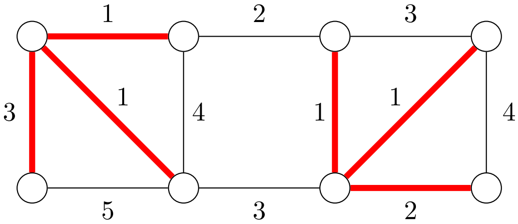


Min weighted Edge Cover Problem

Input: A Graph $G = (V, E, w)$ with $w(e) \geq 0$ for all $e \in E$.

Solution: A subset $S \subseteq E$ which covers all vertices in V.

Output: Minimizing $w(S) = \sum_{e \in S} w(e)$.



Max-Min Edge Cover (Weighted Upper Edge Cover)

Input: A Graph $G = (V, E, w)$ with $w(e) \geq 0$ for all $e \in E$.

Solution: An inclusion-wise minimal edge cover $S \subseteq E$.

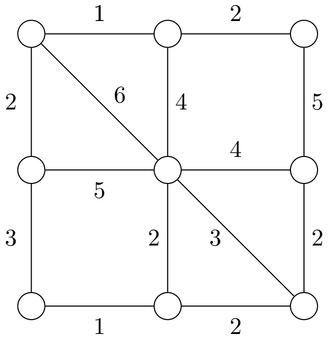
Output: Maximizing $w(S) = \sum_{e \in S} w(e)$.

Max-Min Edge Cover (Weighted Upper Edge Cover)

Input: A Graph $G = (V, E, w)$ with $w(e) \geq 0$ for all $e \in E$.

Solution: An inclusion-wise minimal edge cover $S \subseteq E$.

Output: Maximizing $w(S) = \sum_{e \in S} w(e)$.

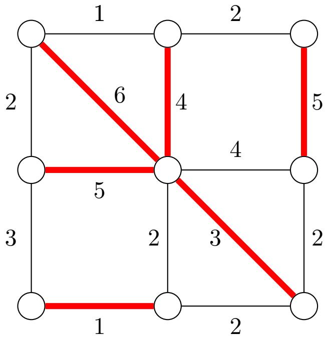


Max-Min Edge Cover (Weighted Upper Edge Cover)

Input: A Graph $G = (V, E, w)$ with $w(e) \geq 0$ for all $e \in E$.

Solution: An inclusion-wise minimal edge cover $S \subseteq E$.

Output: Maximizing $w(S) = \sum_{e \in S} w(e)$.



Max Spanning Star Forest Problem

Input: A Graph $G = (V, E, w)$ with $w(e) \geq 0$ for all $e \in E$.

Solution: A spanning star forest $S = \{S_1, \ldots, S_p\} \subseteq E$.

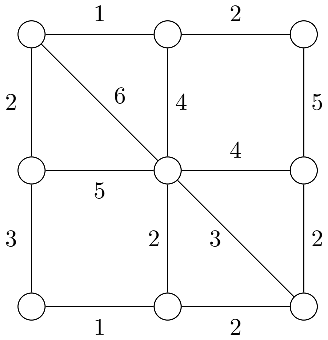
Output: Maximizing $w(S) = \sum_{e \in S} w(e) = \sum_{i=1}^{p} \sum_{e \in S_i} w(e)$

Max Spanning Star Forest Problem

Input: A Graph $G = (V, E, w)$ with $w(e) \geq 0$ for all $e \in E$.

Solution: A spanning star forest $S = \{S_1, \ldots, S_p\} \subseteq E$.

Output: Maximizing $w(S) = \sum_{e \in S} w(e) = \sum_{i=1}^{p} \sum_{e \in S_i} w(e)$

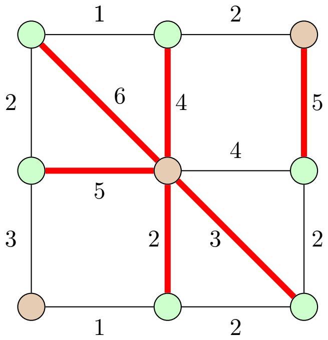


Max Spanning Star Forest Problem

Input: A Graph $G = (V, E, w)$ with $w(e) \geq 0$ for all $e \in E$.

Solution: A spanning star forest $S = \{S_1, \ldots, S_p\} \subseteq E$.

Output: Maximizing $w(S) = \sum_{e \in S} w(e) = \sum_{i=1}^{p} \sum_{e \in S_i} w(e)$

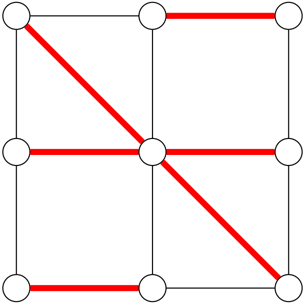


Upper Edge Cover vs. Max Spanning Star Forest

Upper Edge Cover vs. Max Spanning Star Forest

Unweighted version

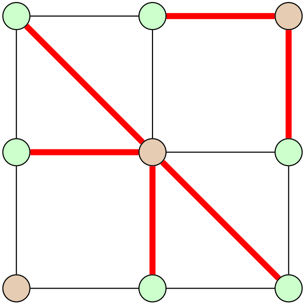
- Any minimal edge cover of graph G is a spanning star forest of G.



Upper Edge Cover vs. Max Spanning Star Forest

Unweighted version

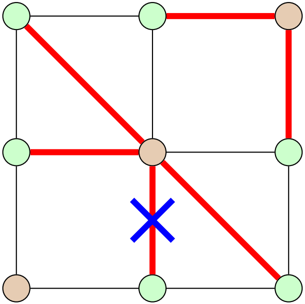
- Any spanning star forest can be converted into a spanning star forest without trivial stars with same value.



Upper Edge Cover vs. Max Spanning Star Forest

Unweighted version

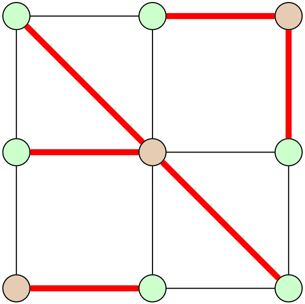
- Any spanning star forest can be converted into a spanning star forest without trivial stars with same value.



Upper Edge Cover vs. Max Spanning Star Forest

Unweighted version

- Any spanning star forest can be converted into a spanning star forest without trivial stars with same value.



Upper Edge Cover vs. Max Spanning Star Forest

Unweighted version

- Any minimal edge cover of graph G is a spanning star forest of G.
- Any spanning star forest can be converted into a spanning star forest without trivial stars with same value.
- Upper Edge Cover and Max SSF are completely equivalent, even in approximation.

Upper Edge Cover vs. Max Spanning Star Forest

Weighted version

- Any minimal edge cover of graph G is a spanning star forest of G.

Upper Edge Cover vs. Max Spanning Star Forest

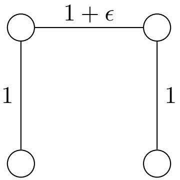
Weighted version

- Any minimal edge cover of graph G is a spanning star forest of G.
  - $opt_{WSSF}(G, w) \geq opt_{WUEC}(G, w)$.

Upper Edge Cover vs. Max Spanning Star Forest

Weighted version

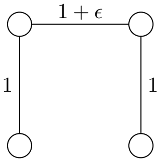
- Any minimal edge cover of graph G is a spanning star forest of G.
  - $opt_{WSSF}(G, w) \geq opt_{WUEC}(G, w)$.



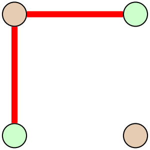
Upper Edge Cover vs. Max Spanning Star Forest

Weighted version

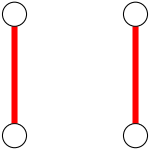
- Any minimal edge cover of graph G is a spanning star forest of G.
  - $opt_{WSSF}(G, w) \geq opt_{WUEC}(G, w)$.



WSSF



WUEC

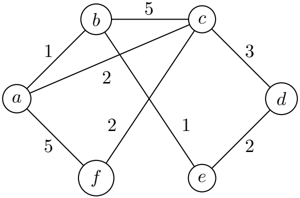


Weighted UEC in Complete graphs

- MAX WSSF in general graphs is equivalent to WEIGHTED UEC in complete graphs.

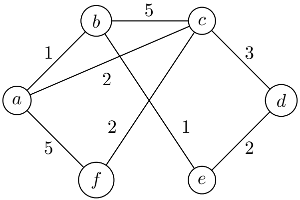
Weighted UEC in Complete graphs

- Max WSSF in general graphs is equivalent to Weighted UEC in complete graphs.

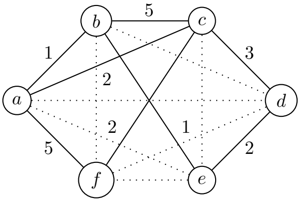


Weighted UEC in Complete graphs

- Max WSSF in general graphs is equivalent to Weighted UEC in complete graphs.



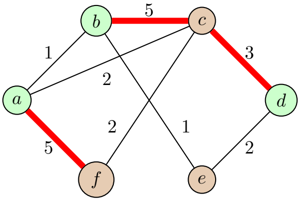
Max WSSF



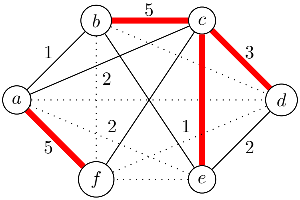
Weighted UEC

Weighted UEC in Complete graphs

- Max WSSF in general graphs is equivalent to Weighted UEC in complete graphs.



Max WSSF



Weighted UEC

Weighted UEC in Complete graphs

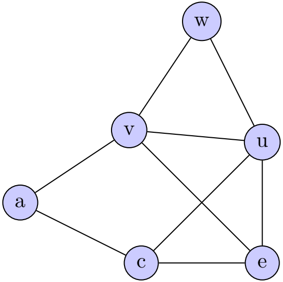
- MAX WSSF in general graphs is equivalent to WEIGHTED UEC in complete graphs.
  - WEIGHTED UEC is 0.5-apx in complete graphs. [D. Chakrabarty et al. 2010]
  - WEIGHTED UEC is $\frac{10}{11}$ in-apx in complete graphs unless P=NP. [C. T. Nguyen et al. 2008]

Weighted UEC in Bipartite graphs

- WEIGHTED UEC in bipartite graphs is as hard as MAX IS in general graphs.

Weighted UEC in Bipartite graphs

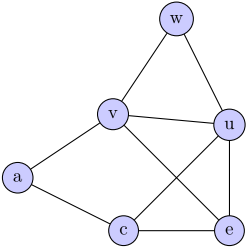
- Weighted UEC in bipartite graphs is as hard as Max IS in general graphs.



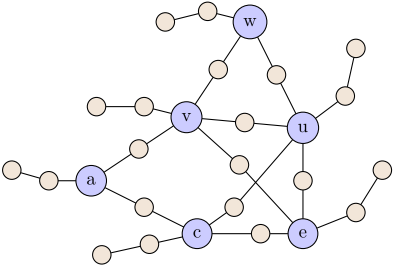
Max IS

Weighted UEC in Bipartite graphs

- Weighted UEC in bipartite graphs is as hard as Max IS in general graphs.



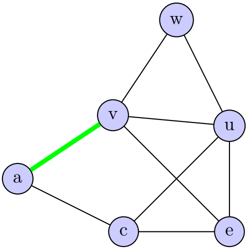
Max IS



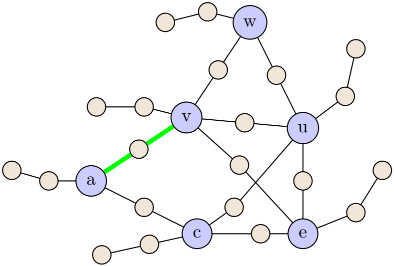
Weighted UEC

Weighted UEC in Bipartite graphs

- Weighted UEC in bipartite graphs is as hard as Max IS in general graphs.



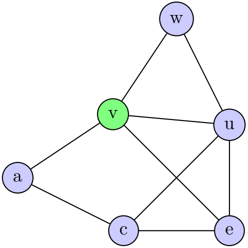
Max IS



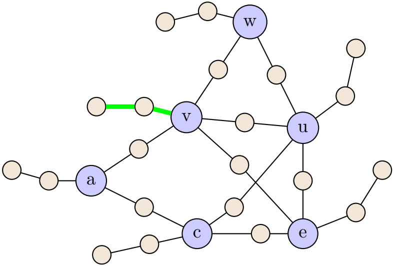
Weighted UEC

Weighted UEC in Bipartite graphs

- Weighted UEC in bipartite graphs is as hard as Max IS in general graphs.



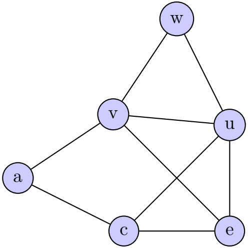
Max IS



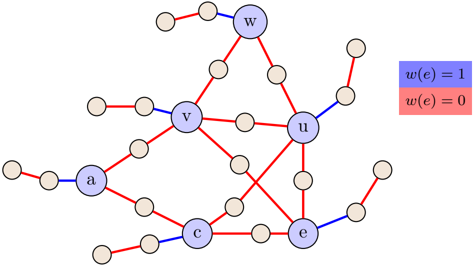
Weighted UEC

Weighted UEC in Bipartite graphs

- Weighted UEC in bipartite graphs is as hard as Max IS in general graphs.



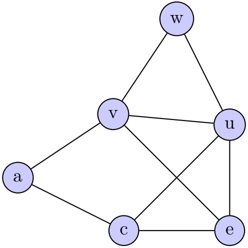
Max IS



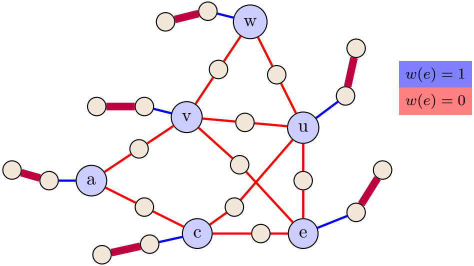
Weighted UEC

Weighted UEC in Bipartite graphs

- Weighted UEC in bipartite graphs is as hard as Max IS in general graphs.



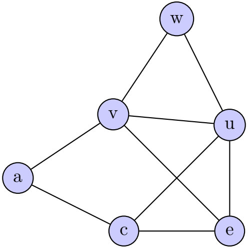
Max IS



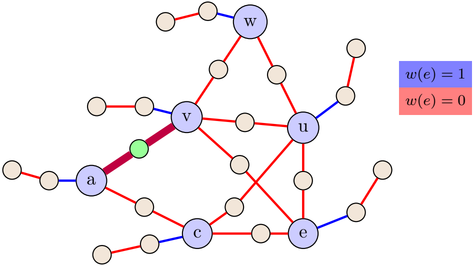
Weighted UEC

Weighted UEC in Bipartite graphs

- WEIGHTED UEC in bipartite graphs is as hard as MAX IS in general graphs.



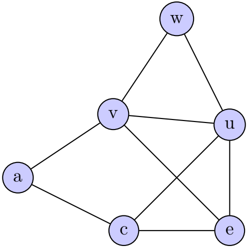
Max IS



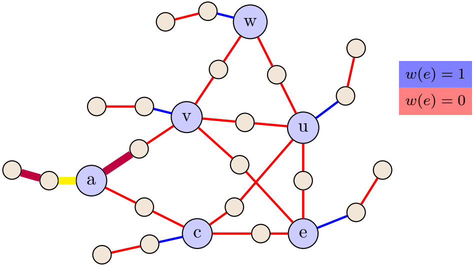
Weighted UEC

Weighted UEC in Bipartite graphs

- **Weighted UEC** in bipartite graphs is as hard as **Max IS** in general graphs.



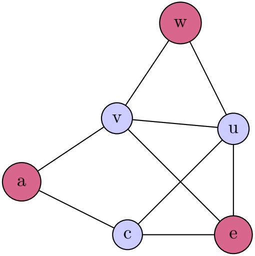
Max IS



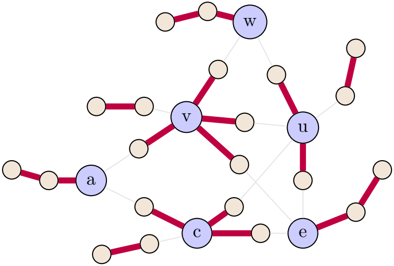
Weighted UEC

Weighted UEC in Bipartite graphs

- Weighted UEC in bipartite graphs is as hard as Max IS in general graphs.



Max IS



Weighted UEC

Weighted UEC in Bipartite graphs

- WEIGHTED UEC in bipartite graphs is as hard as MAX IS in general graphs.
  - For any $\varepsilon > 0$, WEIGHTED UEC in bipartite graphs with n vertices is not $O(n^{\varepsilon-\frac{1}{2}})$ -apx.
  - For any $\varepsilon > 0$, it is hard to approximate WEIGHTED UEC in bipartite graphs og max. degree Δ within a factor $O(\Delta^{\varepsilon-1})$.

Weighted UEC in Graphs of max. degree Δ

WEIGHTED UEC is APX-complete in graphs of max. degree Δ.

Weighted UEC in Graphs of max. degree Δ

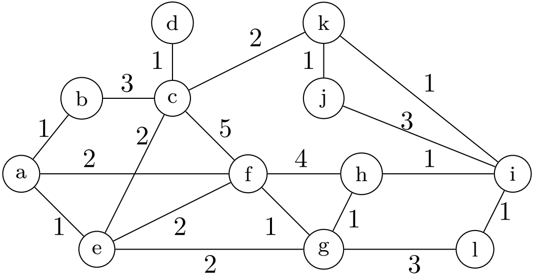
WEIGHTED UEC is APX-complete in graphs of max. degree Δ.

- For any $\varepsilon > 0$, it is hard to approximate WEIGHTED UEC in bipartite graphs of max. degree Δ within a factor $O(\Delta^{\varepsilon-1})$.
- WEIGHTED UEC is $\frac{1}{2.\Delta}$ approximable in graphs of max. degree Δ .

Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- Let $G = (V, E, w)$ is a graph of max. degree Δ.

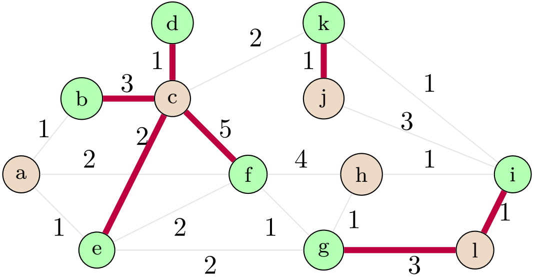


$G = (V, E, w)$ with $\Delta = 5$

Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- Suppose $S = \{S_1, ..., S_k\}$ is an approximation solution of Max WSSF which contains all pendant edges.

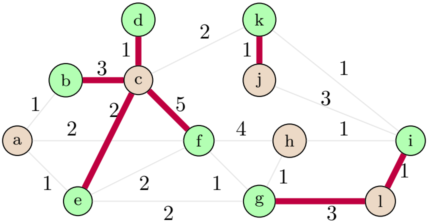


$G = (V, E, w)$ with $\Delta = 5$

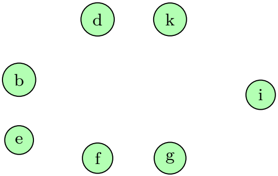
Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- We build a vertex-weighted graph G' with max. degree $\Delta - 1$.



$G = (V, E, w)$ with $\Delta = 5$

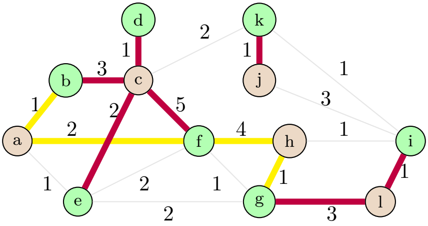


$G' = (V', E', w')$ with $\Delta' \leq 4$

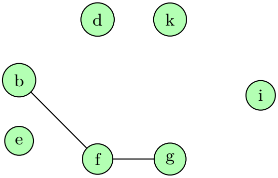
Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- We build a vertex-weighted graph G' with max. degree $\Delta - 1$.



$G = (V, E, w)$ with $\Delta = 5$

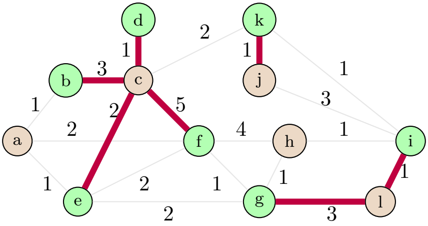


$G' = (V', E', w')$ with $\Delta' \leq 4$

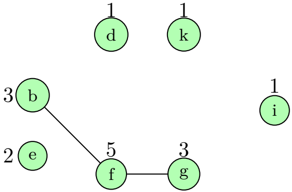
Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- We build a vertex-weighted graph G' with max. degree $\Delta - 1$.



$G = (V, E, w)$ with $\Delta = 5$

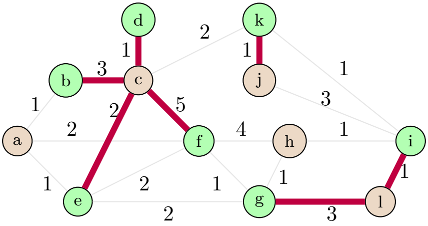


$G' = (V', E', w')$ with $\Delta' \leq 4$

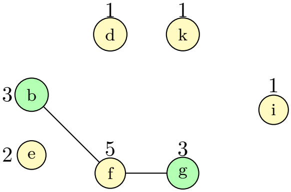
Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- Find Max Weighted IS in G' greedily by weights.



$G = (V, E, w)$ with $\Delta = 5$

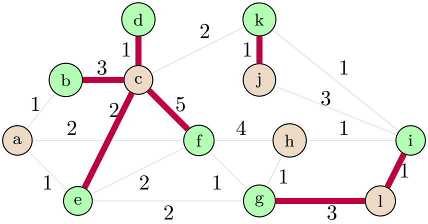


$G' = (V', E', w')$ with $\Delta' \leq 4$

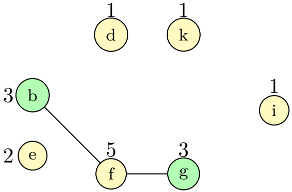
Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- Find Max Weighted IS in G' greedily by weights.



$G = (V, E, w)$ with $\Delta = 5$



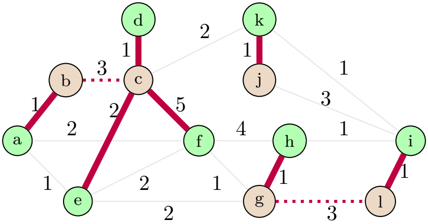
$G' = (V', E', w')$ with $\Delta' \leq 4$

$w(I') \geq \frac{1}{\Delta'+1} \sum_{v \in V'} w(v) \geq \frac{1}{\Delta} \sum_{v \in V'} w(v)$

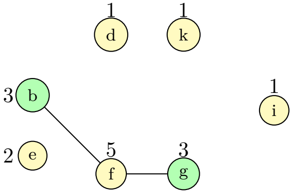
Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- Find Max Weighted IS in G' greedily by weights.



$G = (V, E, w)$ with $\Delta = 5$



$G' = (V', E', w')$ with $\Delta' \leq 4$

$w(S') \geq w(I') \geq \frac{1}{\Delta} \sum_{e \in S} w(e) = \frac{1}{\Delta} w(S)$

$w(I') \geq \frac{1}{\Delta'+1} \sum_{v \in V'} w(v) \geq \frac{1}{\Delta} \sum_{v \in V'} w(v)$

Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- Max Extended Spanning Star Forest is $\frac{1}{2}$-approximable. [Khoshkhah et al. 2018]

Weighted UEC in graphs of max. degree Δ

Approximation algorithm

- Max Extended Spanning Star Forest is $\frac{1}{2}$ approximable. [Khoshkhah et al. 2018]

$$w(S) \geq \frac{1}{2} opt_{ESSF}(G) \geq \frac{1}{2} opt_{WUEC}(G)$$

$$w(S') \geq \frac{1}{2.\Delta} opt_{WUEC}$$

Weighted UEC in k-trees

A k-tree defined inductively as follows:

- A K_{k+1} is a k-tree.
- If G is a k-tree, then adding a new vertex which has exactly k neighbors in G such that these $k+1$ vertices induce a K_{k+1}, forms a k-tree.

Weighted UEC in k-trees

Weighted UEC is APX-complete in k-trees.

Weighted UEC in k-trees

WEIGHTED UEC is APX-complete in k-trees.

- WEIGHTED UEC is not approximable within $\frac{10}{11}$ in complete weighted graphs.

Weighted UEC in k-trees

WEIGHTED UEC is APX-complete in k-trees.

- WEIGHTED UEC is not approximable within $\frac{10}{11}$ in complete weighted graphs.
- WEIGHTED UEC is $\frac{k-1}{2(k+1)}$ approximable in k-trees.

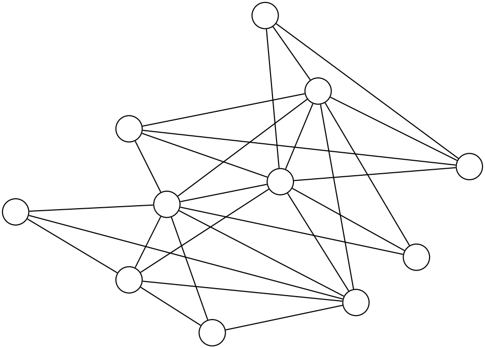
Weighted UEC in k-trees

Approximation algorithm

Weighted UEC in k-trees

Approximation algorithm

- Let $G = (V, E, w)$ is a k-tree.

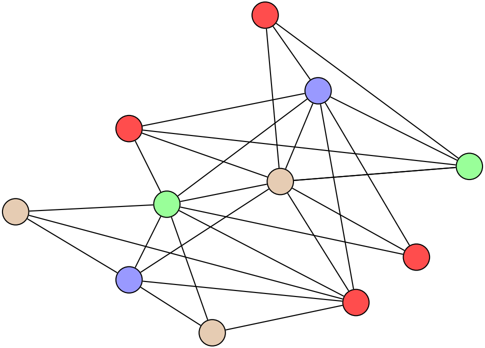


3-trees, $G = (V, E, w)$

Weighted UEC in k-trees

Approximation algorithm

- Any k-tree can be colored greedily by $k+1$ colors in linear time.

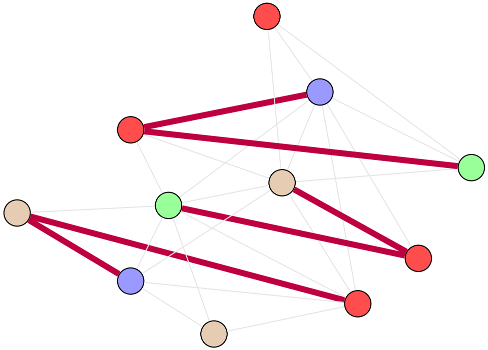


3-trees, $G = (V, E, w)$

Weighted UEC in k-trees

Approximation algorithm

- Suppose $S = \{S_1, ..., S_r\}$ is a nice spanning star forest of the k-tree.



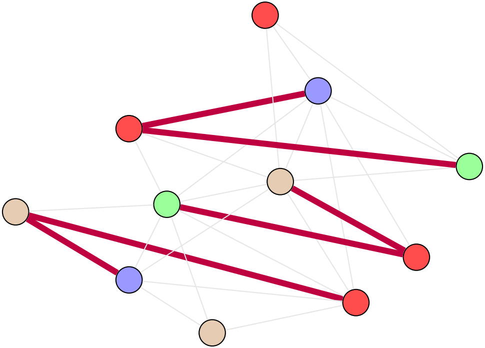
3-trees, $G = (V, E, w)$

Weighted UEC in k-trees

Approximation algorithm

- Modify spanning star forest S to obtain an edge cover S'.

Property. let c_1, c_2 be two distinct colors of coloring C. For each trivial star v in S, let $V' = \{v \in N(v) : C(v) \in \{c_1, c_2\}\}$, then for any $u \in V'$, $S' = (S \setminus \{uc\}) \cup \{uv\}$ is a new solution.

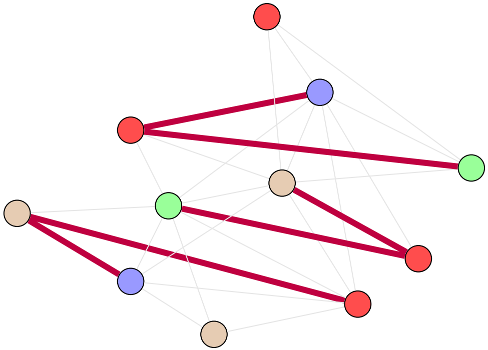


3-trees, $G = (V, E, w)$

Weighted UEC in k-trees

Approximation algorithm

For each color c_i, let $E_{c_i} = \{cv \in S : C(v) = c_i\}$.

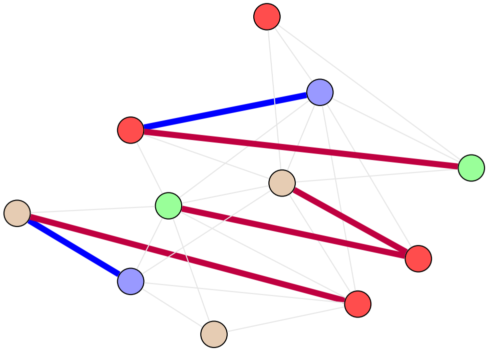


3-trees, $G = (V, E, w)$

Weighted UEC in k-trees

Approximation algorithm

For each color c_i, let $E_{c_i} = \{cv \in S : C(v) = c_i\}$.

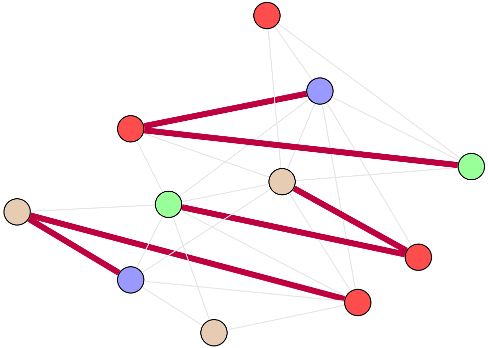


3-trees, $G = (V, E, w)$

Weighted UEC in k-trees

Approximation algorithm

Suppose for two distinct colors c_1, c_2, $w(E_{c_1} \cup E_{c_2}) = \min\{w(E_{c_i} \cup E_{c_j}) : i, j \in \{1, ..., k+1\}\}$, then for each trivial vertex v in S, remove an edge $uc \in \{E_{c_1} \cup E_{c_2}\}$ from S and add the edge uv to S.

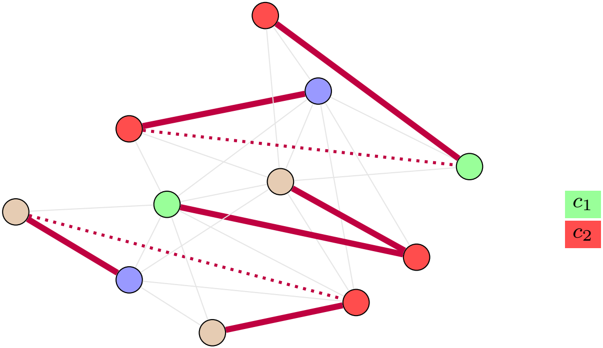


3-trees, $G = (V, E, w)$

Weighted UEC in k-trees

Approximation algorithm

Suppose for two distinct colors c_1, c_2, $w(E_{c_1} \cup E_{c_2}) = \min\{w(E_{c_i} \cup E_{c_j}) : i, j \in \{1, ..., k+1\}\}$, then for each trivial vertex v in S, remove an edge $uc \in \{E_{c_1} \cup E_{c_2}\}$ from S and add the edge uv to S.



3-trees, $G = (V, E, w)$

Weighted UEC in k-trees

Approximation algorithm

$$w(E_{c_i} \cup E_{c_j}) \leq \frac{2}{k+1} w(E(S))$$

$$w(S') \geq w(E(S) - (E_{c_1} \cup E_{c_2})) \geq \frac{k-1}{k+1} w(E(S))$$

Weighted UEC in k-trees

Approximation algorithm

$$w(E_{c_i} \cup E_{c_j}) \leq \frac{2}{k+1} w(E(S))$$

$$w(S') \geq w(E(S) - (E_{c_1} \cup E_{c_2})) \geq \frac{k-1}{k+1} w(E(S))$$

- Max WSSF is $\frac{1}{2}$ approximable in k-trees.

Weighted UEC is $\frac{k-1}{2(k+1)}$ approximable

Summary of results

Summary of results

Weighted Upper Edge Cover		
	Negative result	Positive result
Complete graphs	$\frac{10}{11}$ in-apx	a $\frac{1}{2}$ apx-alg

Summary of results

Weighted Upper Edge Cover		
	Negative result	Positive result
Complete graphs	$\frac{10}{11}$ in-apx	a $\frac{1}{2}$ apx-alg
Bipartite graphs	$O(n)^{\varepsilon-\frac{1}{2}}$ in-apx	—

Summary of results

Weighted Upper Edge Cover		
	Negative result	Positive result
Complete graphs	$\frac{10}{11}$ in-apx	a $\frac{1}{2}$ apx-alg
Bipartite graphs	$O(n)^{\epsilon-\frac{1}{2}}$ in-apx	—
Graphs of max degree Δ	$O(\Delta)^{\epsilon-1}$ in-apx	a $\frac{1}{2.\Delta}$ apx-alg

Summary of results

Weighted Upper Edge Cover		
	Negative result	Positive result
Complete graphs	$\frac{10}{11}$ in-apx	a $\frac{1}{2}$ apx-alg
Bipartite graphs	$O(n)^{\epsilon-\frac{1}{2}}$ in-apx	—
Graphs of max degree Δ	$O(\Delta)^{\epsilon-1}$ in-apx	a $\frac{1}{2.\Delta}$ apx-alg
Split graphs	$O(n)^{\epsilon-\frac{1}{2}}$ in-apx	—

Summary of results

Weighted Upper Edge Cover		
	Negative result	Positive result
Complete graphs	$\frac{10}{11}$ in-apx	a $\frac{1}{2}$ apx-alg
Bipartite graphs	$O(n)^{\epsilon-\frac{1}{2}}$ in-apx	—
Graphs of max degree Δ	$O(\Delta)^{\epsilon-1}$ in-apx	a $\frac{1}{2.\Delta}$ apx-alg
Split graphs	$O(n)^{\epsilon-\frac{1}{2}}$ in-apx	—
k-trees	$\frac{259}{260}$ in-apx	a $\frac{k-1}{2(k+1)}$ apx-alg

Thank you for your attention.