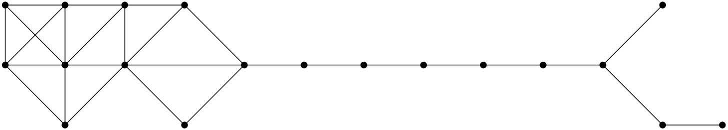
Graph burning and non-uniform k-centers for small treewidth

Matej Lieskovský and **Jiří Sgall**

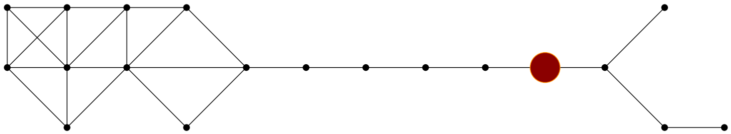
Charles University, Prague

Aussois, May 2022

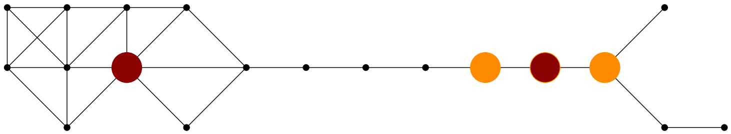
Graph Burning



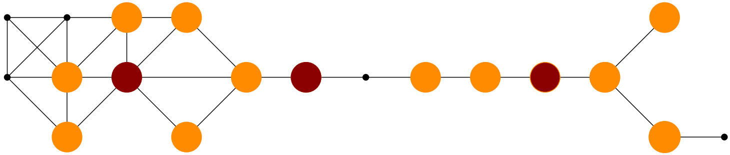
Graph Burning



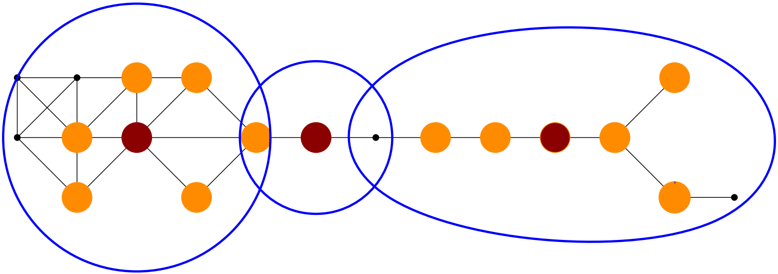
Graph Burning



Graph Burning



Graph Burning



Definition: Graph burning [Bonato *et al.* 2015]

- Input: Graph $G = (V, E)$
- Output: Vertices $u_1, \ldots, u_g \in V$, so that for each $v \in V$ there exists i with $d(v, u_i) \leq g - i - 1$
- Objective: Minimize g

Non-uniform k-center

Definition: k-center [Hakimi 1964]

- Input: Graph $G = (V, E)$, k
- Output: Vertices $u_1, \ldots, u_k \in V$, and r so that for each $v \in V$ exist an i with $d(v, u_i) \le r$
- Objective: Minimize r

Non-uniform k-center

Definition: k-center [Hakimi 1964]

- Input: Graph $G = (V, E)$, k
- Output: Vertices $u_1, \ldots, u_k \in V$, and r so that for each $v \in V$ exist an i with $d(v, u_i) \leq r$
- Objective: Minimize r

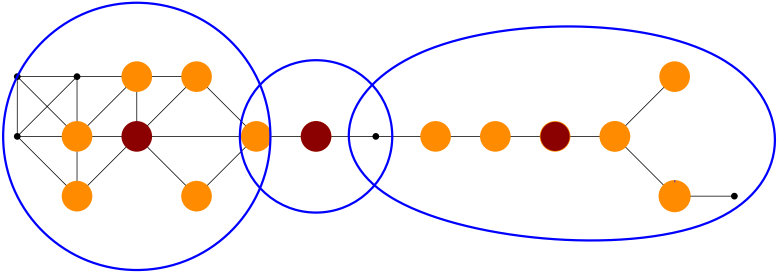
Definition: Non-uniform k-center [Chakrabarty *et al.* 2020]

- Input: Graph $G = (V, E)$, k, radii $r_1, \ldots, r_k$
- Output: Vertices $u_1, \ldots, u_k \in V$, so that for each $v \in V$ exist an i with $d(v, u_i) \leq r_i$
- Objective: Test feasibility

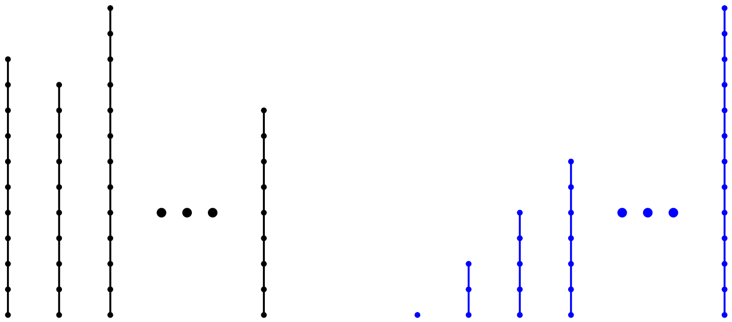
Non-uniform k-center

Definition: Non-uniform k-center

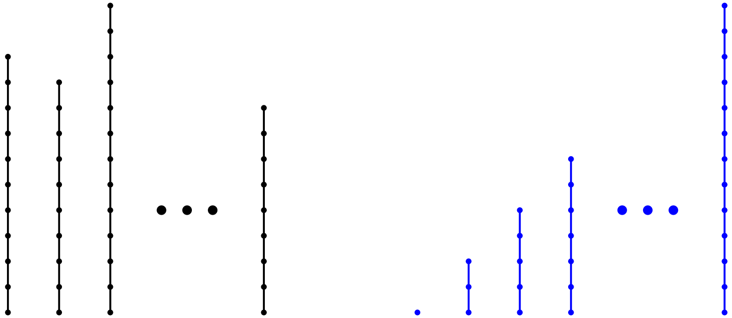
- Input: Graph $G = (V, E)$, k, radii $r_1, \ldots, r_k$
- Output: Vertices $u_1, \ldots, u_k \in V$, so that for each $v \in V$ exist an i with $d(v, u_i) \leq r_i$



Paths and linear forests



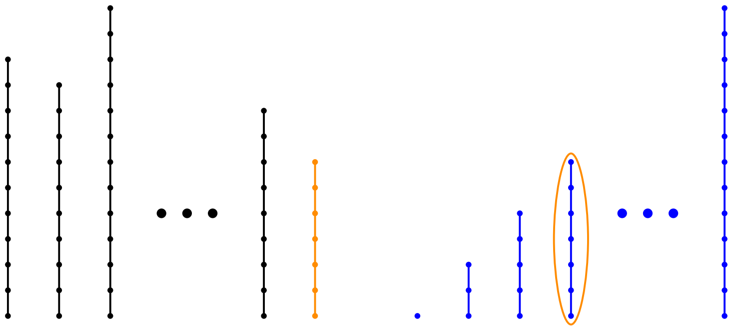
Paths and linear forests



For sets of disjoint paths, Graph burning

- is NP-hard, and [Bessy *et al.* 2017]
- has a PTAS. [Bonato, Kamali 2019]

Paths and linear forests



For sets of disjoint paths, Graph burning

- is NP-hard, and [Bessy *et al.* 2017]
- has a PTAS. [Bonato, Kamali 2019]

Our results

Graph burning

PTAS for trees and graphs of bounded treewidth.

Our results

Graph burning

PTAS for trees and graphs of bounded treewidth.

Non-uniform *k*-center

XP algorithm parametrized by both the number of distinct radii and treewidth.

Our results

Graph burning

PTAS for trees and graphs of bounded treewidth.

Non-uniform *k*-center

XP algorithm parametrized by both the number of distinct radii and treewidth.

Reduction:
Use radii $\varepsilon \cdot g$, $2\varepsilon \cdot g$, ..., with $\varepsilon \cdot g$ centers using each one.

Algorithm for trees

Dynamic programming over a tree

Process the tree bottom up.
For each vertex, remember the set of possible configurations each containing:

- The number of centers of each type used in the subtree
- The maximal coverage that can be achieved, which is
  - either $+a$: the centers in the subtree can cover nodes up to the distance a from the processed vertex
  - or $-a$: some nodes in the subtree up to the distance a from the processed vertex need to be covered by a center outside of the tree

Algorithm for trees

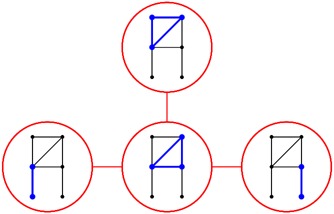
Dynamic programming over a tree

Process the tree bottom up.
For each vertex, remember the set of possible configurations each containing:

- The number of centers of each type used in the subtree
- The maximal coverage that can be achieved, which is
  - either $+a$: the centers in the subtree can cover nodes up to the distance a from the processed vertex
  - or $-a$: some nodes in the subtree up to the distance a from the processed vertex need to be covered by a center outside of the tree

Can be generalized to arbitrary edge lengths

Treewidth: Introduction



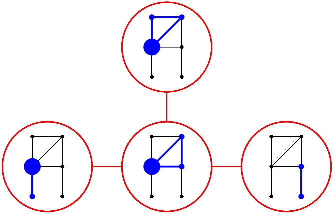
Definition of treewidth

A tree decomposition of G is a tree of bags such that

- each bag contains a subgraph of G,
- each edge appears in some bag, and
- each vertex is contained in a connected subset of bags.

The treewidth is the maximal number of vertices in a bag, minus 1.

Treewidth: Introduction



Definition of treewidth

A tree decomposition of G is a tree of bags such that

- each bag contains a subgraph of G,
- each edge appears in some bag, and
- each vertex is contained in a connected subset of bags.

The treewidth is the maximal number of vertices in a bag, minus 1.

Algorithm for bounded treewidth

Tidy decompositions

If a vertex v is covered by a center at u, any vertex on any shortest path from v to u is also covered by the same center.

Algorithm for bounded treewidth

Tidy decompositions

If a vertex v is covered by a center at u, any vertex on any shortest path from v to u is also covered by the same center.

Dynamic programming over a tree of bags

Process the tree bottom up.
For each bag, remember the set of possible configurations each containing:

- The number of centers of each type used in the already processed subtree
- For each vertex in the bag, the center serving this vertex

Algorithm for bounded treewidth

Tidy decompositions

If a vertex v is covered by a center at u, any vertex on any shortest path from v to u is also covered by the same center.

Dynamic programming over a tree of bags

Process the tree bottom up.
For each bag, remember the set of possible configurations each containing:

- The number of centers of each type used in the already processed subtree
- For each vertex in the bag, the center serving this vertex

Tidy decompositions can be verified locally on each edge.

Non-uniform k-center

k-center [Hochbaum, Shmoys 1985]

2-approximation, this is optimal.

Non-uniform k-center

k-center [Hochbaum, Shmoys 1985]

2-approximation, this is optimal.

Greedy algorithm: Greedily pick vertices with distances $> 2r$ from each other.

- If at most k vertices picked, these are centers with radius $2r$.
- If $> k$ vertices found, each of them needs a different center of radius r, so it is not a feasible instance.

Non-uniform k-center

k-center [Hochbaum, Shmoys 1985]

2-approximation, this is optimal.

Greedy algorithm: Greedily pick vertices with distances $> 2r$ from each other.

- If at most k vertices picked, these are centers with radius $2r$.
- If $> k$ vertices found, each of them needs a different center of radius r, so it is not a feasible instance.

Non-uniform k-center with two radii [Chakrabarty *et al.* 2020]

- 2-approximation if one radius is 0 (aka "outliers")
- $(1 + \sqrt{2})$-approximation otherwise

Graph burning

Greedy algorithm is a 3-approximation [Bessy *et al.* 2017]

Pick vertices greedily with distances $> 2g$ from each other

- If at most g vertices picked, these centers burn the graph in time $3g$.
- If $> g$ vertices found, each of them needs a different center of radius $\leq g$, so it is not a feasible instance.

Open problem

Find a better approximation for general graphs.