Parameterized Approximation Algorithms for some Location Problems in Graphs

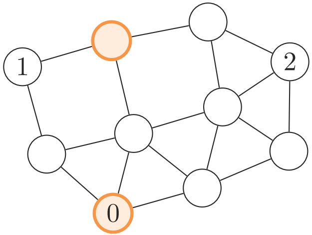
Arne Leitert and Feodor F. Dragan

r-Domination and p-Cencter

r-Domination Problem

(Connected) r-Domination Problem

For a given graph $G = (V, E)$ and given function $r\colon V \to \mathbb{N}$, determine a (connected) vertex set D with minimum cardinality such that, for each vertex v, $d(v, D) \le r(v)$.

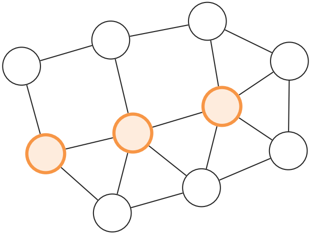


Optimal r-domination.

p-Center Problem

(Connected) p-Center Problem

For a given graph $G = (V, E)$ and given integer p, determine a (connected) vertex set D with $|D| \leq p$ such that $\max_{v \in V} d(v, D)$ is minimal.



Optimal connected 3-center.

r-Domination vs. p-Center

Two versions of the same problem.

r-Domination

- Given: Maximal distance.
- Find: Best cardinality.

p-Center

- Given: Maximal cardinality.
- Find: Lowest maximum distance.

An algorithm for one problem gives an algorithm for the other problem with low computational overhead.

Approximation for r-Domination

Theorem [Chlebík, Chlebíková 2008]

Under reasonable assumptions, the r-Domination problem cannot be approximated within a factor of $(1 - \varepsilon) \ln n$ in polynomial time. (Even for very restricted graphs).

Our Approach

- Do not approximate cardinality, approximate range of r.
- Goal: Find $(r + \phi)$-dominating set not larger than the optimal set.
- An $(r + \phi)$-dominating set gives an $+\phi$-approximation for the p-Center problem.

Existing Result

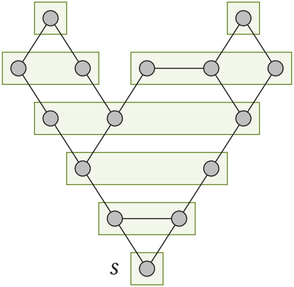
- $(r + 2\delta)$-dominating set in polynomial time for δ-hyperbolic graphs [Chepoi, Estellon 2007]

Layering Partition

Layering Partition

Layering Partition [Brandstädt et al. 1999; Chepoi, Dragan 2000]

- Distance layers for a given vertex s
- Partition each layer: u and v are in the same cluster if they are connected by a path only using the same or upper layers
- Computable in linear time



Δ denotes max. distance (in G) of two vertices in a cluster.

Breadth of Real-Life Networks [Abu-Ata, Dragan 2014]

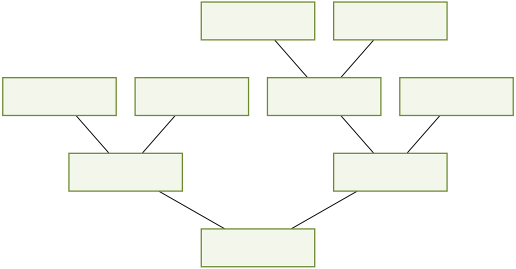
Graph	Δ
PPI	5
Yeast	4
DutchElite	6
EPA	4
EVA	5
California	4
Erdös	2
Routeview	4
Homo release 3.2.99	3
AS_Caida_20071105	3
Dimes 3/2010	2
Aqualab 12/2007- 09/2008	3
AS_Caida_20120601	3
itdk0304	6
DBLB-coauth	7
Amazon	12

Algorithm

General Approach

Idea

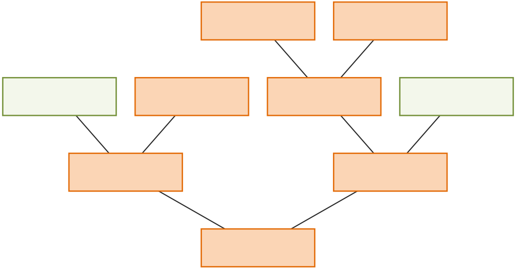
- Compute layering partition $\mathcal{T}$ for graph G.



General Approach

Idea

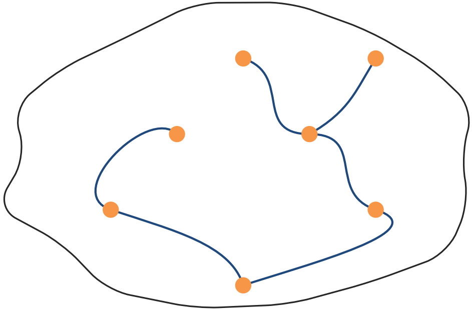
- Compute layering partition $\mathcal{T}$ for graph G.
- Solve problem for $\mathcal{T}$.



General Approach

Idea

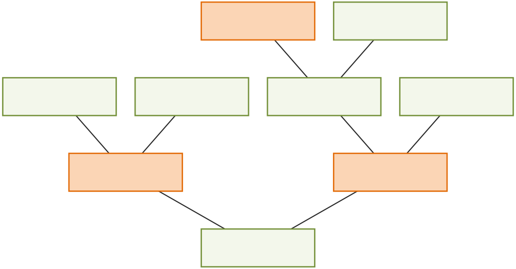
- Compute layering partition $\mathcal{T}$ for graph G.
- Solve problem for $\mathcal{T}$.
- Use solution for $\mathcal{T}$ to compute solution for underlying graph G.



r-Domination (non-connected)

Solving r-Domination for $\mathcal{T}$

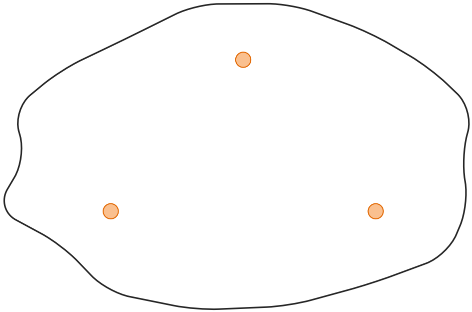
- Set $r(C) = \min_{v \in C} r(v)$ for each cluster C of $\mathcal{T}$.
- Find minimum r-dominating set $\mathcal{S}$ for $\mathcal{T}$.



r-Domination (non-connected)

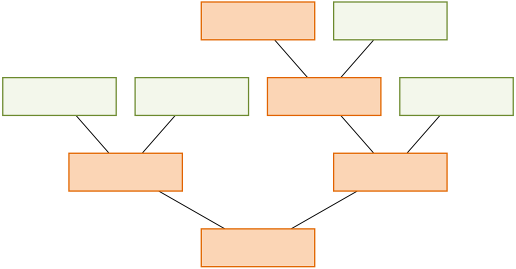
Compute solution for G

- For each cluster $C \in \mathcal{S}$, pick a vertex $v \in C$ and add v into a set D.
- D is an $(r + \Delta)$-dominating set for G.
- Total runtime: linear



Solving Connected r-Domination for $\mathcal{T}$

- Set $r(C) = \min_{v \in C} r(v)$ for each cluster C of $\mathcal{T}$.
- Find minimum connected r-dominating set (i. e., a subtree) T_r for $\mathcal{T}$.
- Useful: $|T_r| \leq |D_r|$
  (D_r is *unknown* optimal con. r-dom. set for G)



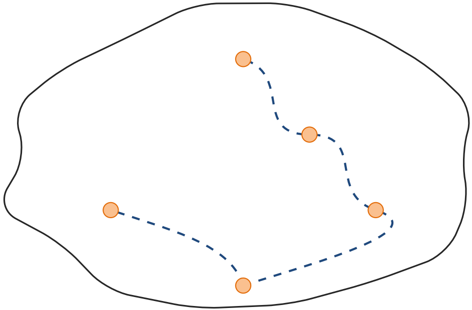
Connected r-Domination

Compute solution for G?

- For each cluster $C \in \mathcal{S}$, pick a vertex $v \in C$ and add v into a set D.

Problems

- How to ensure connectedness?
- How do we ensure cardinality constraints?

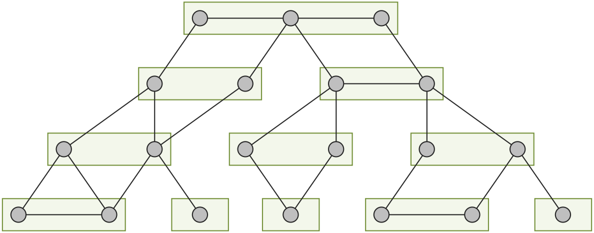


Idea

- Construct $(r + \delta)$-dominating subtree T_δ of $\mathcal{T}$ for some $\delta \in \mathbb{N}$.
- Construct small enough vertex set S_δ of G intersecting all clusters of T_δ
- Try different values for δ until $|S_\delta| \leq |T_r|$ and, thus, $|S_\delta| \leq |D_r|$.

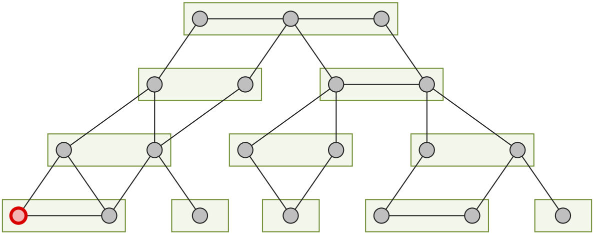
Constructing S_δ – Set of Shortest Paths $\mathcal{P}$

Construct T_δ



Constructing S_δ – Set of Shortest Paths $\mathcal{P}$

Pick a vertex v in an unmarked leaf C (excluding the root) of T_δ

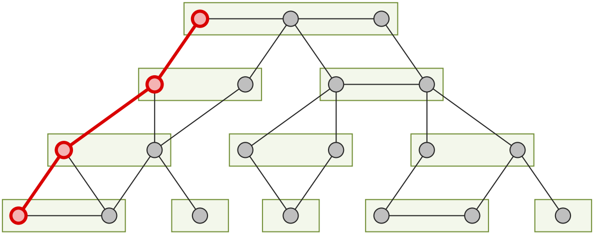


Constructing S_δ – Set of Shortest Paths $\mathcal{P}$

Find the highest unmarked ancestor C' of C and a shortest path P from v to a vertex $v' \in C'$.

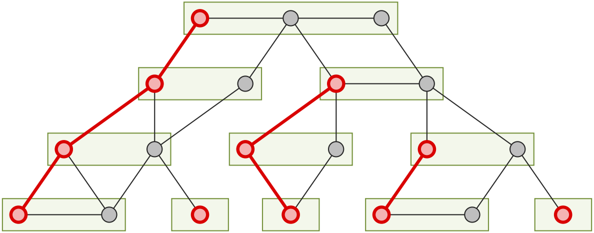
Add P to a set of paths $\mathcal{P}$.

Mark all clusters intersected by P.



Constructing S_δ – Set of Shortest Paths $\mathcal{P}$

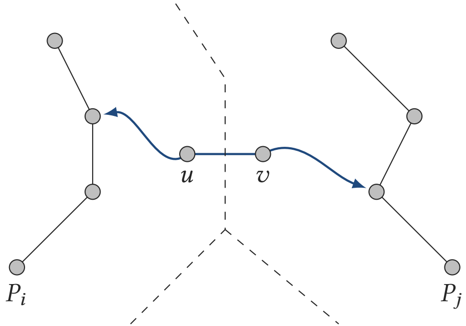
Repeat for each leaf of T_δ.



Constructing S_δ – Connect Paths in $\mathcal{P}$

Run BFS starting simultanisly from all $P \in \mathcal{P}$. Gives a partition $\mathcal{V} = \{V_1, V_2, \ldots\}$ of V.

Connect paths in $\mathcal{P}$ similar to KRUSKAL's MST algorithm based on edges uv with $u \in V_i$ and $v \in V_j$.



Connected r-Domination – Finding best δ

One-Sided Binary Search

- Start with $\delta = 0$. Then, $\delta = 1$, $\delta = 2$, $\delta = 4$, $\delta = 8$, ... until $|S_\delta| \leq |T_r|$.
- Next, classical binary search between last values of δ.
- If $|S_\delta| \leq |T_r|$, decrease δ.
  Otherwise, increase δ.

Result

- Connected $(r + \Delta + \delta)$-dominating set S_δ with $\delta \leq \Delta$ (i. e., $(r + 2\Delta)$-dom. set)
- $+2\Delta$-Approximation for Connected p-Center problem
- Runtime: $\mathcal{O}\Big(m\,\alpha(n)\log\Delta\Big)$

Thank you!