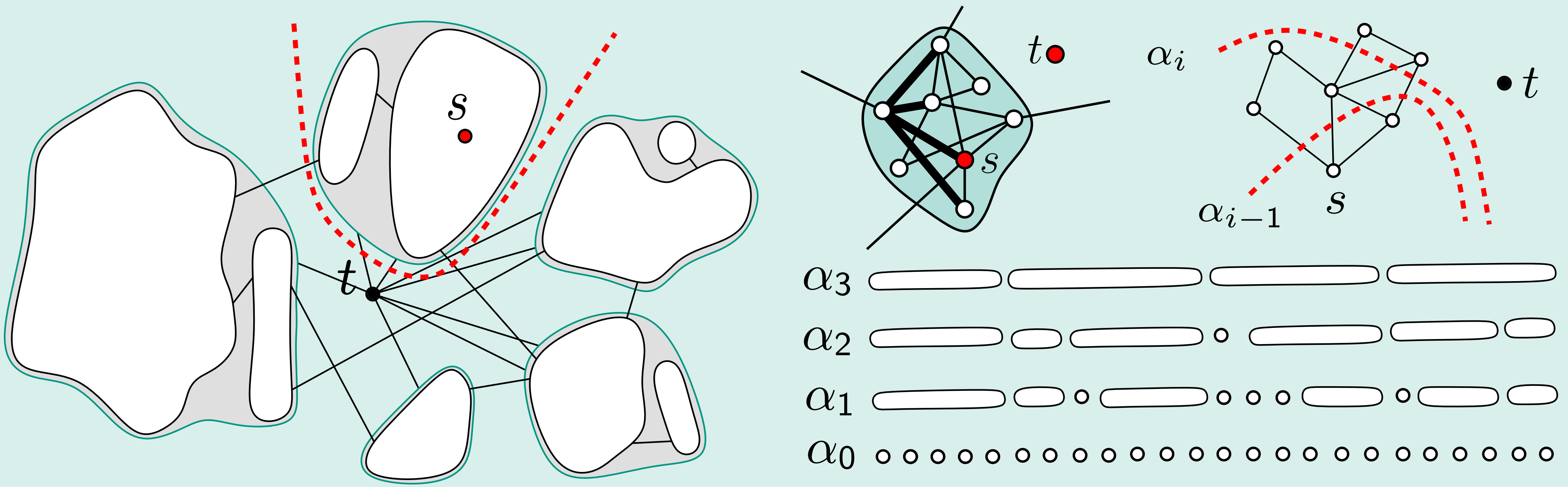
Complete Hierarchical Cut-Clustering: A Case Study on Modularity and Expansion

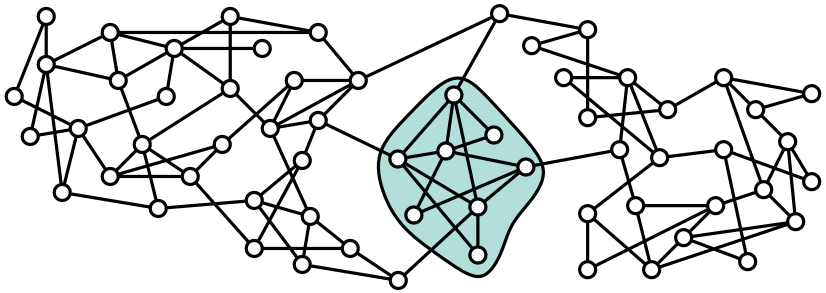
DIMACS 2012
Michael Hamann, Tanja Hartmann, Dorothea Wagner | February 2012

Institute of Theoretical Informatics, Prof. Dr. Dorothea Wagner

Karlsruhe Institute of Technology (KIT)



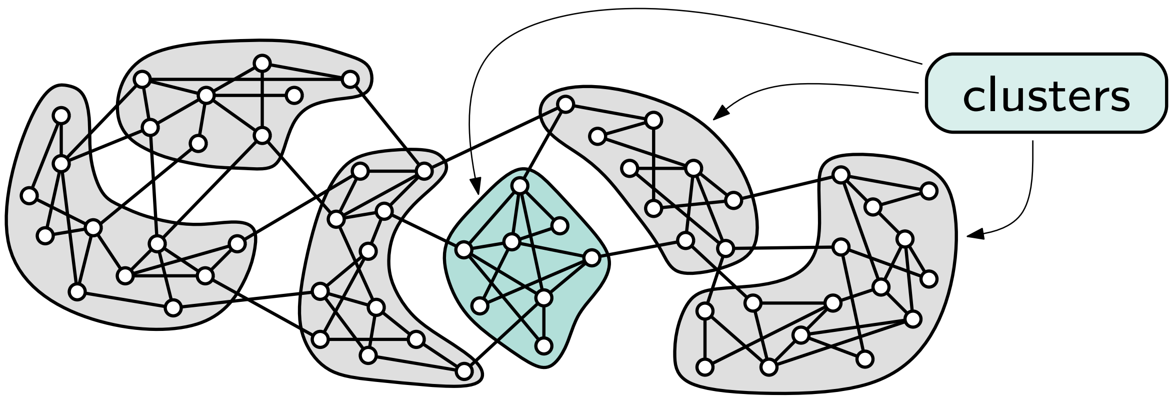
Introduction



The informal aim of graph clustering:

- (weighted) graph with a particular edge structure
- identify subsets of vertices that are significantly related
- weak external relations $\rightarrow$ more significance

Introduction



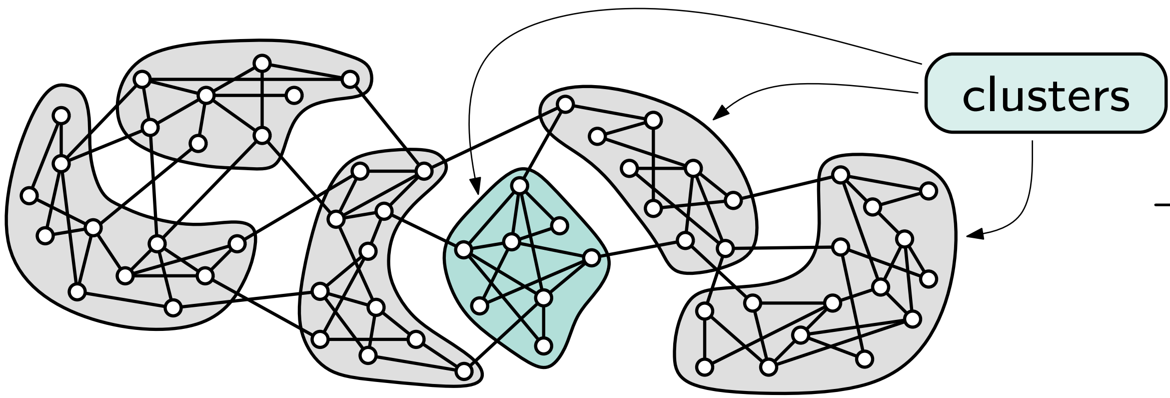
The informal aim of graph clustering:

- (weighted) graph with a particular edge structure
- identify subsets of vertices that are significantly related
- weak external relations $\rightarrow$ more significance

clustering is a decomposition into such subsets,

induced subgraphs are called clusters

Introduction



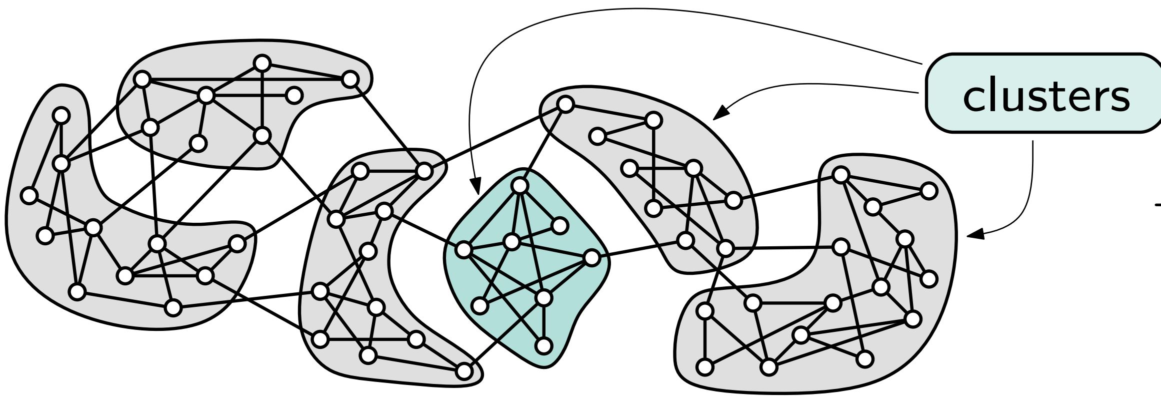
Many formalizations of
significant relations
$\rightarrow$ many measures for quality
$\rightarrow$ many heuristics

The informal aim of graph clustering:

- (weighted) graph with a particular edge structure
- identify subsets of vertices that are significantly related
- weak external relations $\rightarrow$ more significance

clustering is a decomposition into such subsets,
induced subgraphs are called clusters

Introduction



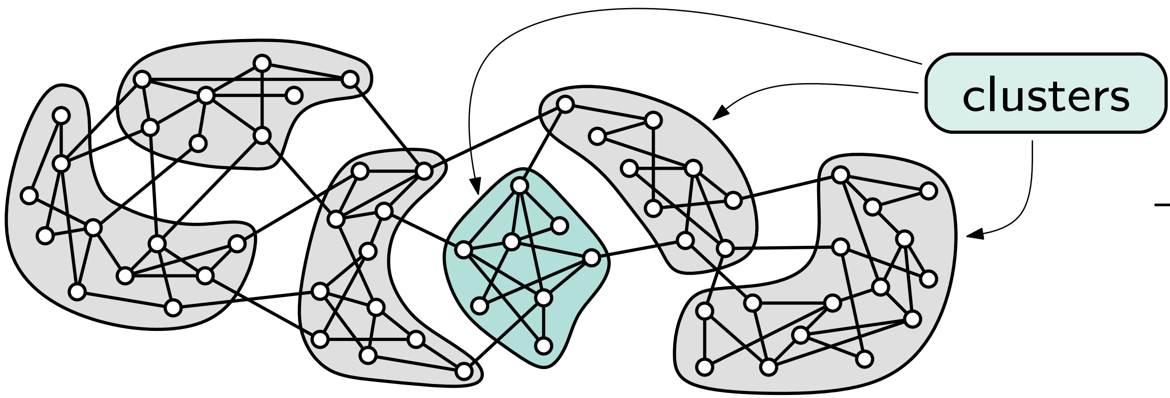
Many formalizations of significant relations
$\rightarrow$ many measures for quality
$\rightarrow$ many heuristics

Modularity

- widely used
- expresses significance of a clustering $\mathcal{C}$ compared to a random clustering

$$\mathcal{M}(\mathcal{C}) := \underbrace{\sum_{C\in\mathcal{C}} \frac{c(E_C)}{c(E)}}_{cov(\mathcal{C})} - \underbrace{\sum_{C\in\mathcal{C}} \frac{\left(\sum_{v\in C} d_c(v)\right)^2}{4c(E)^2}}_{\mathbb{E}(cov(\mathcal{C}))}$$

Introduction



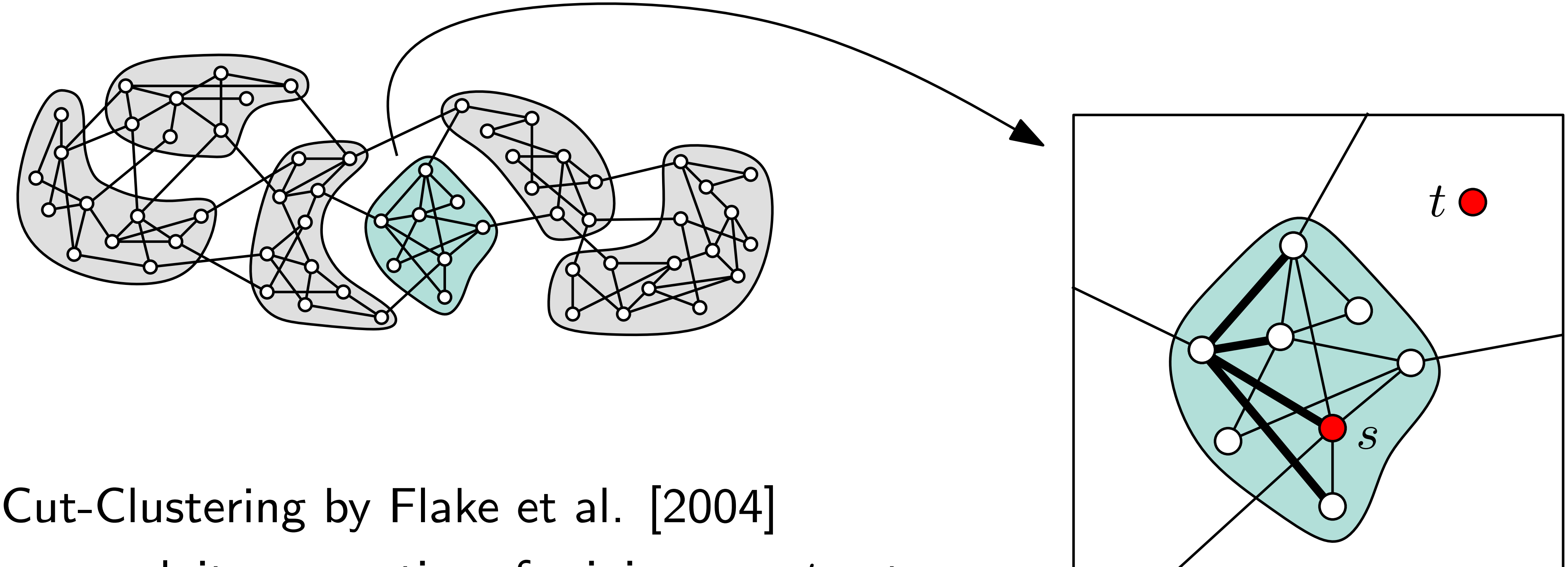
Many formalizations of significant relations
$\rightarrow$ many measures for quality
$\rightarrow$ many heuristics

Cut-Clustering by Flake et al. [2004]

- exploits properties of minimum s-t-cuts
- guarantees a lower bound on intra-cluster expansion (NP-hard)

$$\Psi(\mathcal{C}) := \min_{C\in\mathcal{C}}\{\min_{S\subset C} \frac{c(S,\overline{S})}{\min\{|S|,|\overline{S}|\}}\}$$

Introduction



Cut-Clustering by Flake et al. [2004]

- exploits properties of minimum s-t-cuts
- guarantees a lower bound on intra-cluster expansion (NP-hard)

$$\Psi(\mathcal{C}) := \min_{C \in \mathcal{C}} \{ \min_{S \subset C} \frac{c(S,\overline{S})}{\min\{|S|,|\overline{S}|\}} \}$$

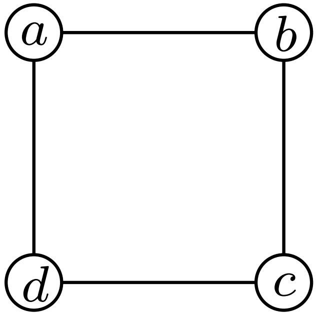
- significance due to clearly indicated membership of vertices to clusters
  - for all $U \subset C \setminus \{s\}$: $c(U,C) \geq c(U, V \setminus C)$
    $\rightarrow$ *source-community* of s

Outline

- The Cut-Clustering Algorithm
  - depends on a parameter steering the coarseness
  - clusterings are hierarchically nested for different parameter values
  - parameter constitutes guarantee on intra-cluster expansion
- Parametric Search Approach
  - returns a complete hierarchy of cut-clusterings
  - simple and efficient (brief running time experiment)
- Modularity Analysis of Cut-Clustering Algotithm
  - compared to a modularity-based greedy approach
- Expansion Analysis of Cut-Clustering Algorithm
  - comparing the guarantee to trivial and non-trivial bounds

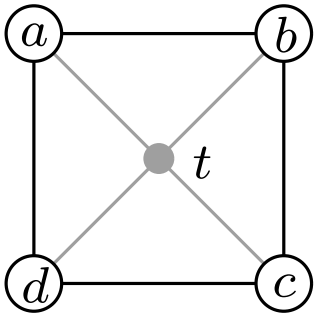
The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion



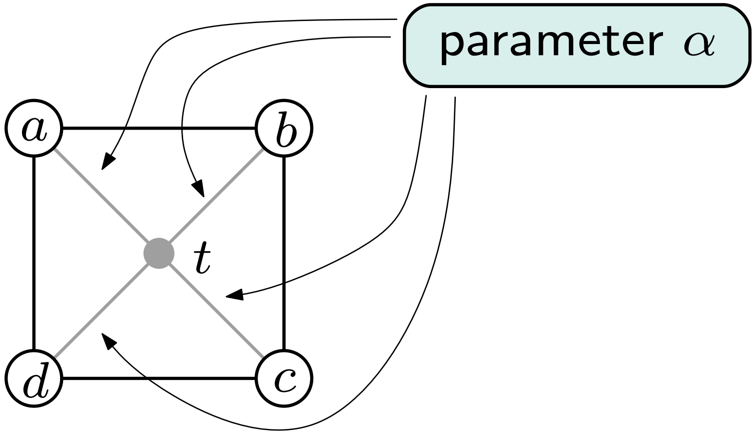
The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion



The Cut-Clustering Algorithm

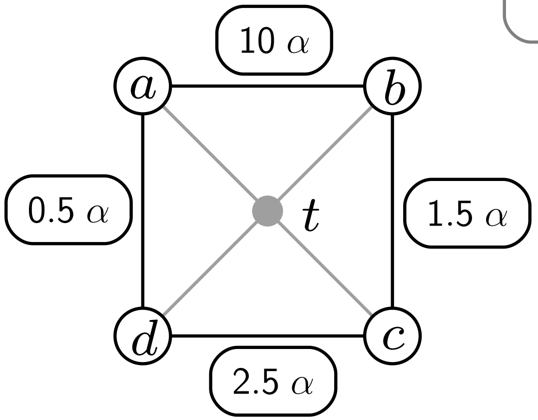
- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$



The Cut-Clustering Algorithm

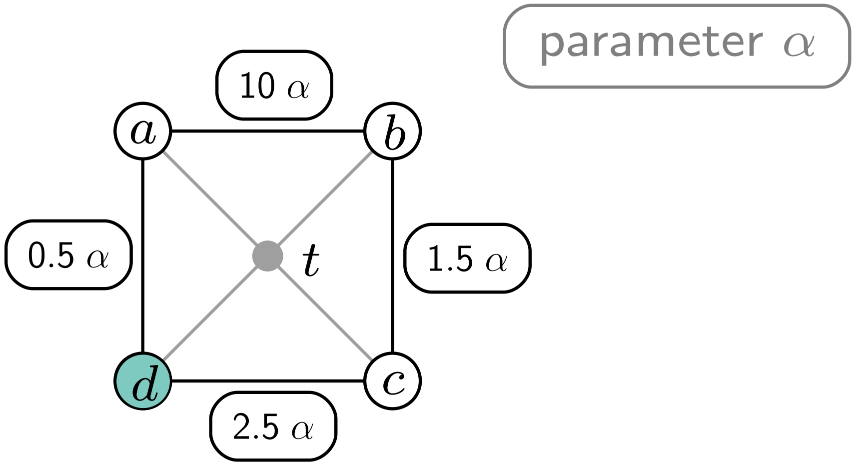
- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$

parameter α



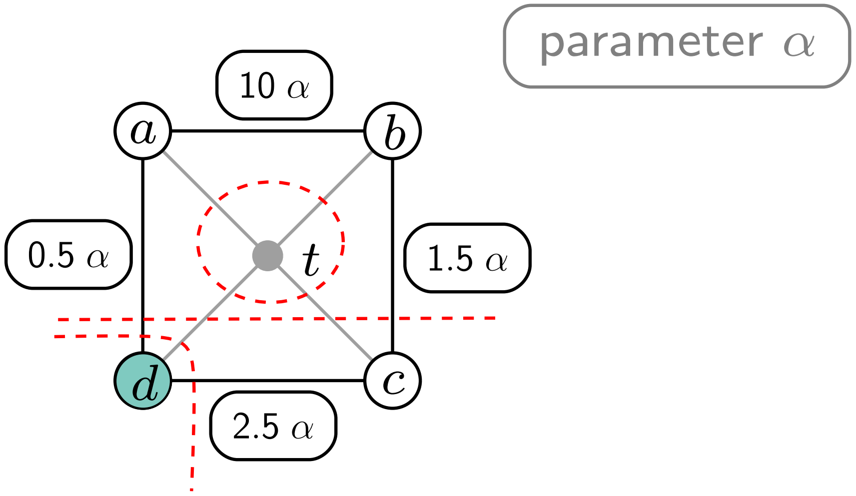
The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$



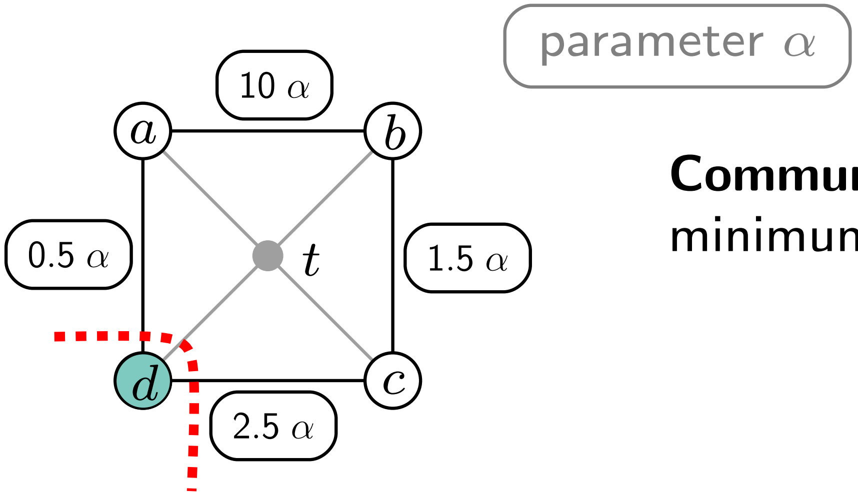
The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\quad \Psi(\mathcal{C}) \geq \alpha$



The Cut-Clustering Algorithm

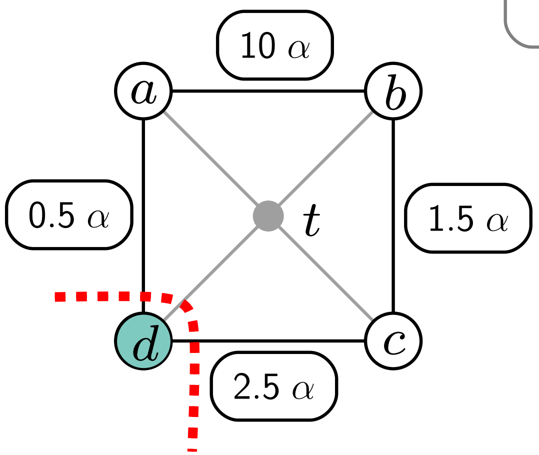
- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$



Community Cut:
minimum s-t-cut with source side of minimum size

The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$

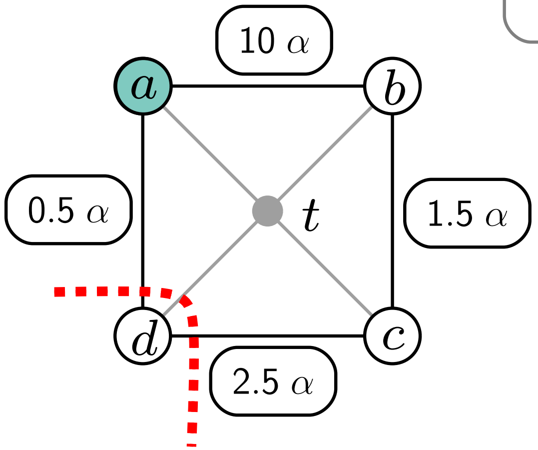


parameter α | unique

Community Cut:
minimum s-t-cut with source side of minimum size

The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$



parameter α

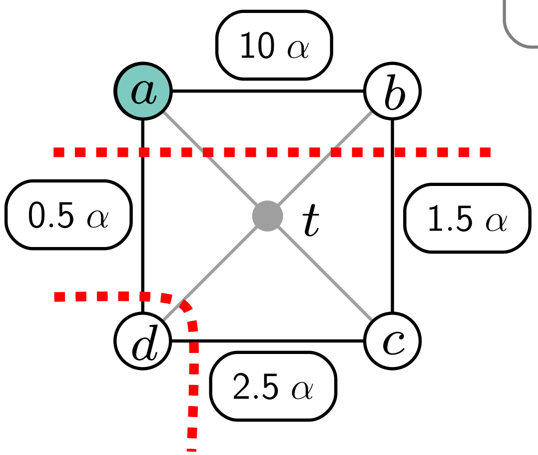
unique

Community Cut:
minimum s-t-cut with source side of minimum size

The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$

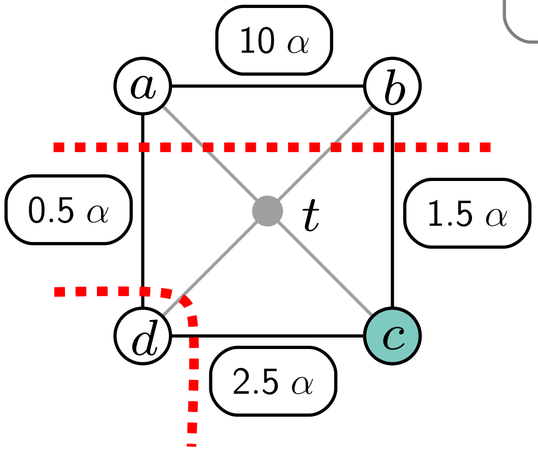
parameter α | unique



Community Cut:
minimum s-t-cut with source side of minimum size

The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$



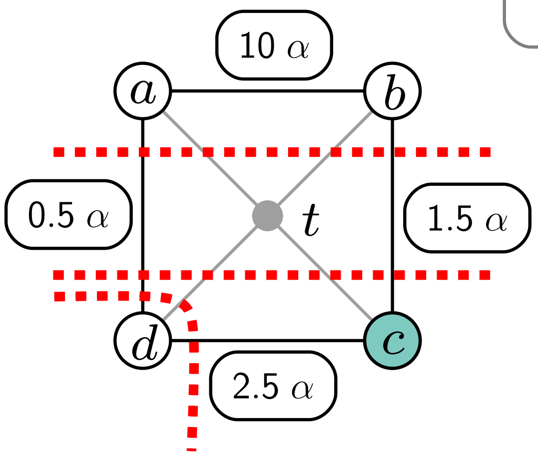
parameter α

unique

Community Cut:
minimum s-t-cut with source side of minimum size

The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\quad \Psi(\mathcal{C}) \geq \alpha$

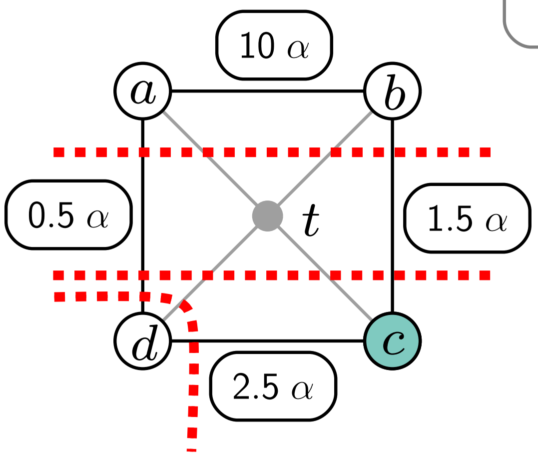


parameter α | unique

Community Cut:
minimum s-t-cut with source side of minimum size

The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$

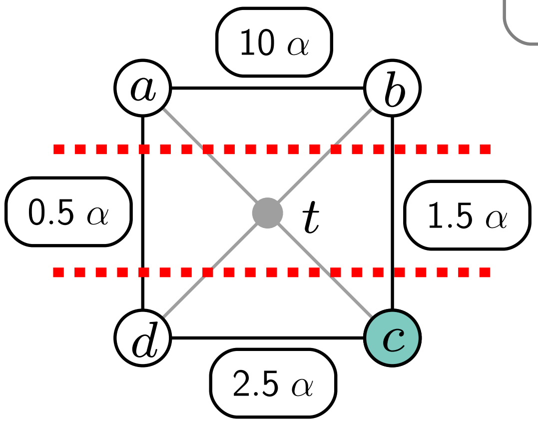


parameter α | unique | non-crossing

Community Cut:
minimum s-t-cut with source side of minimum size

The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$



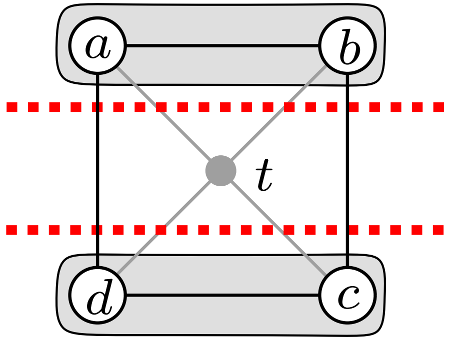
parameter α | unique | non-crossing

Community Cut:
minimum s-t-cut with source side of minimum size

The Cut-Clustering Algorithm

- Flake et al. [2004]
- artificial vertex, artificial edges
- special minimum s-t-cuts (so called *community cuts*)
- clusters are source-communities $\Rightarrow$ good significance of clustering
- guarantees bound on *intra-cluster* expansion $\Psi(\mathcal{C}) \geq \alpha$

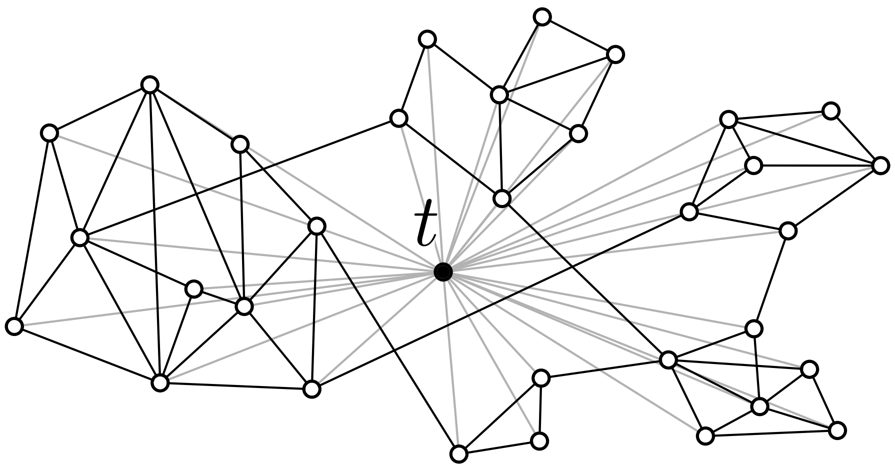
parameter α | unique | non-crossing

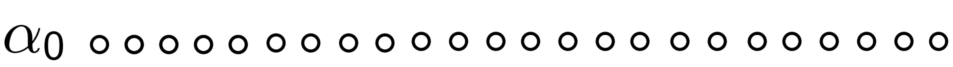


Community Cut:
minimum s-t-cut with source side of minimum size
$\longrightarrow$ Cut-Clustering with two clusters

The Cut-Clustering Algorithm

- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$



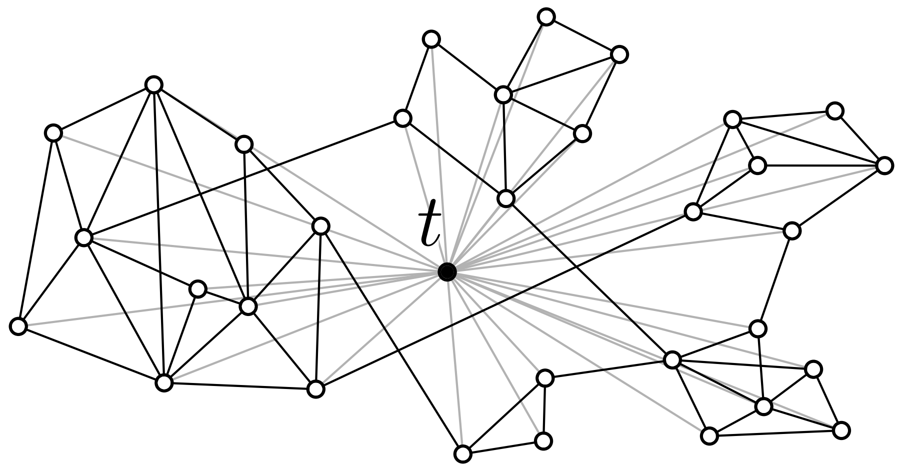


The Cut-Clustering Algorithm

- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

For $i = 1, \ldots, k$ do

- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i

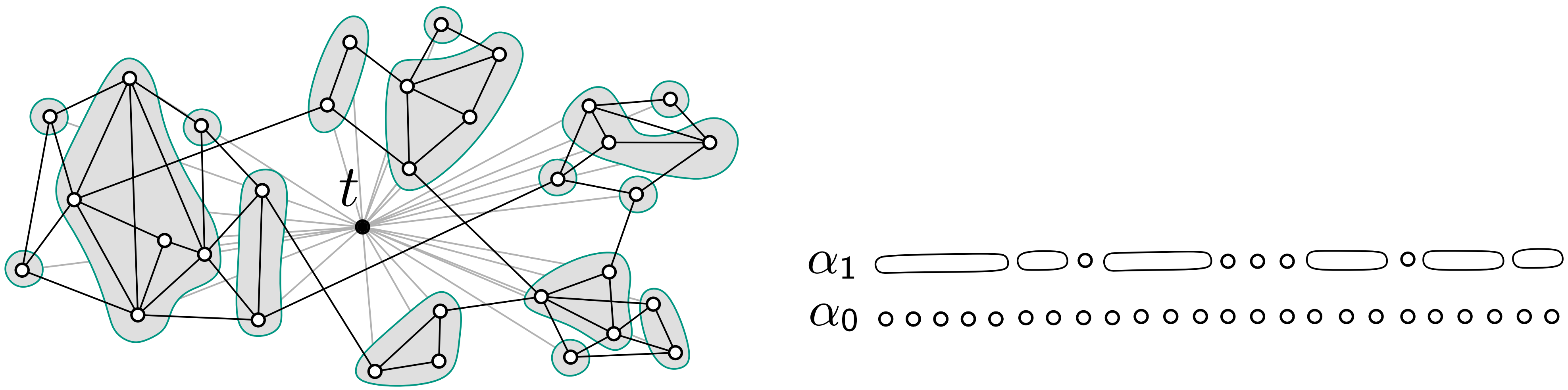


The Cut-Clustering Algorithm

- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

For $i = 1, \ldots, k$ do

- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i

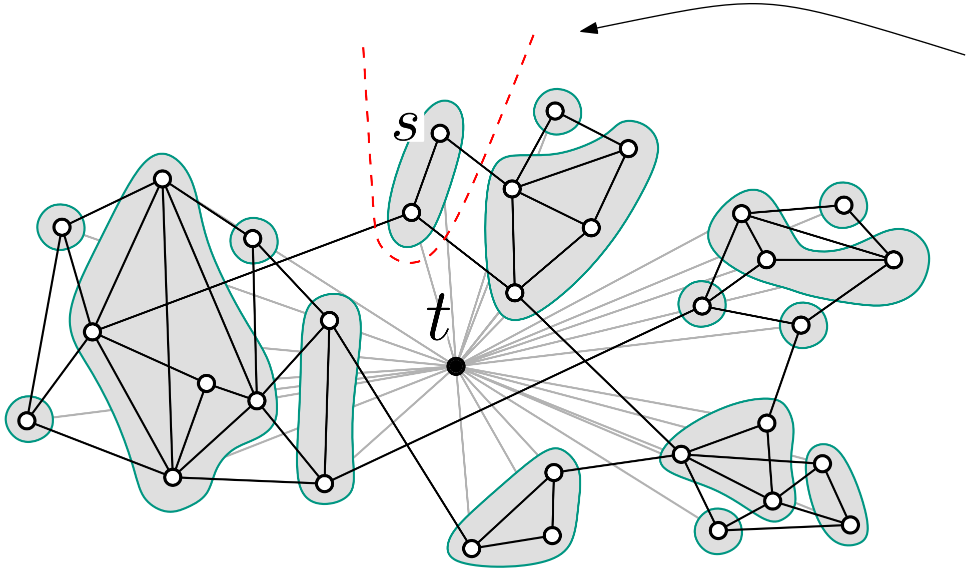


The Cut-Clustering Algorithm

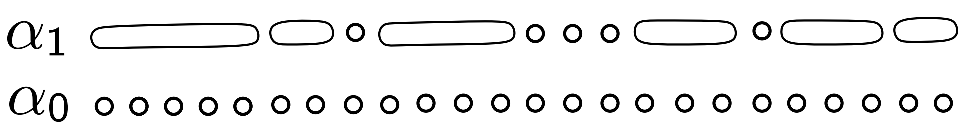
- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

For $i = 1, \ldots, k$ do

- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i



Source Community of s in **underlying** graph

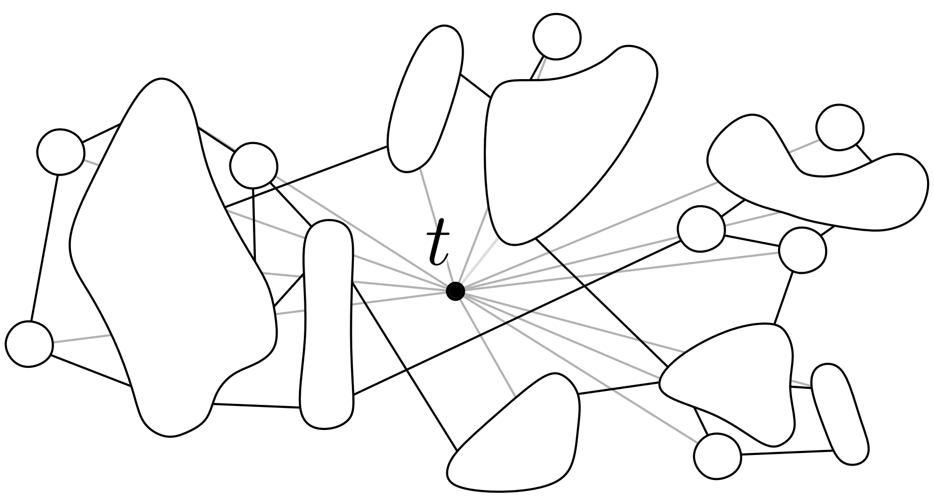


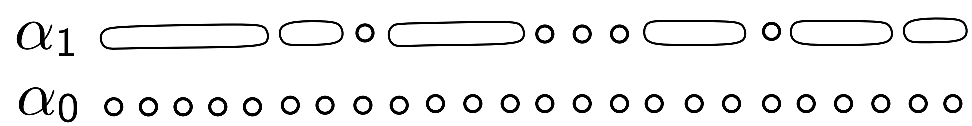
The Cut-Clustering Algorithm

- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

For $i = 1, \ldots, k$ do

- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i



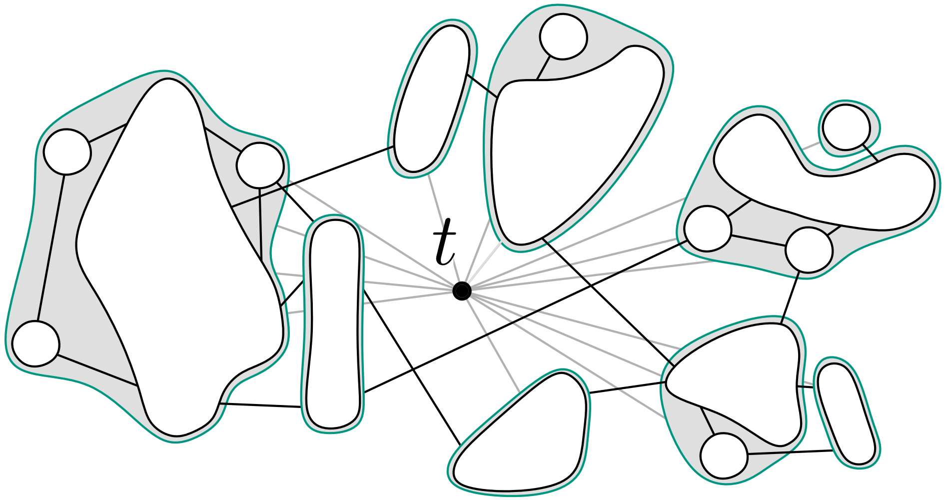


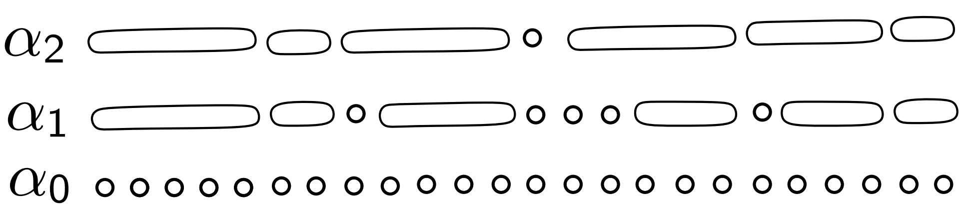
The Cut-Clustering Algorithm

- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

For $i = 1, \ldots, k$ do

- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i



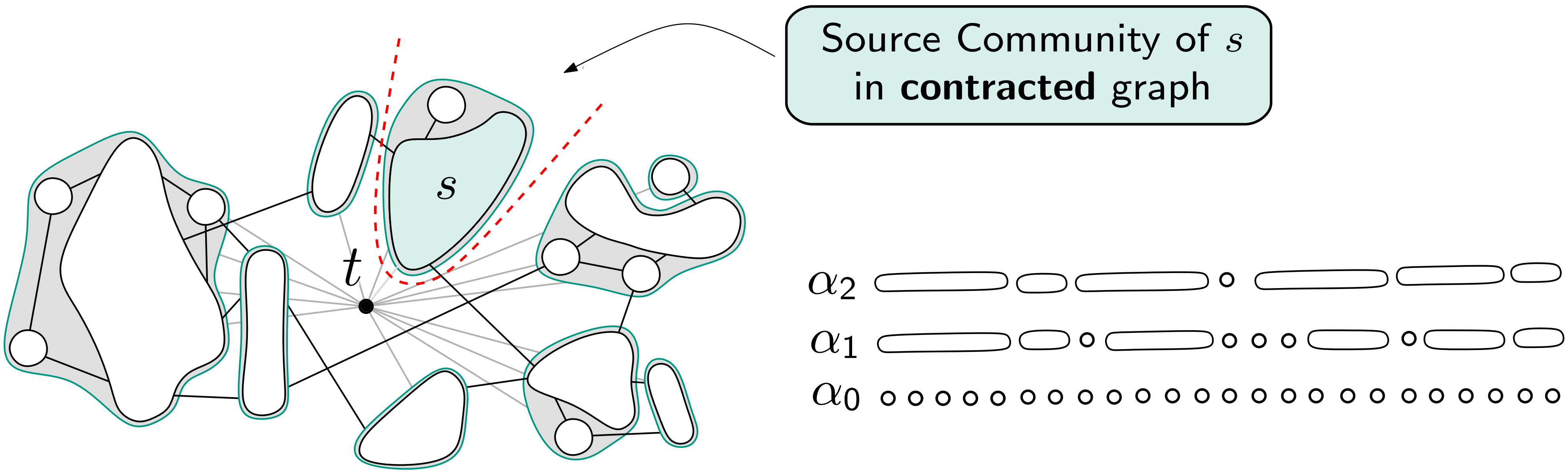


The Cut-Clustering Algorithm

- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

For $i = 1, \ldots, k$ do

- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i

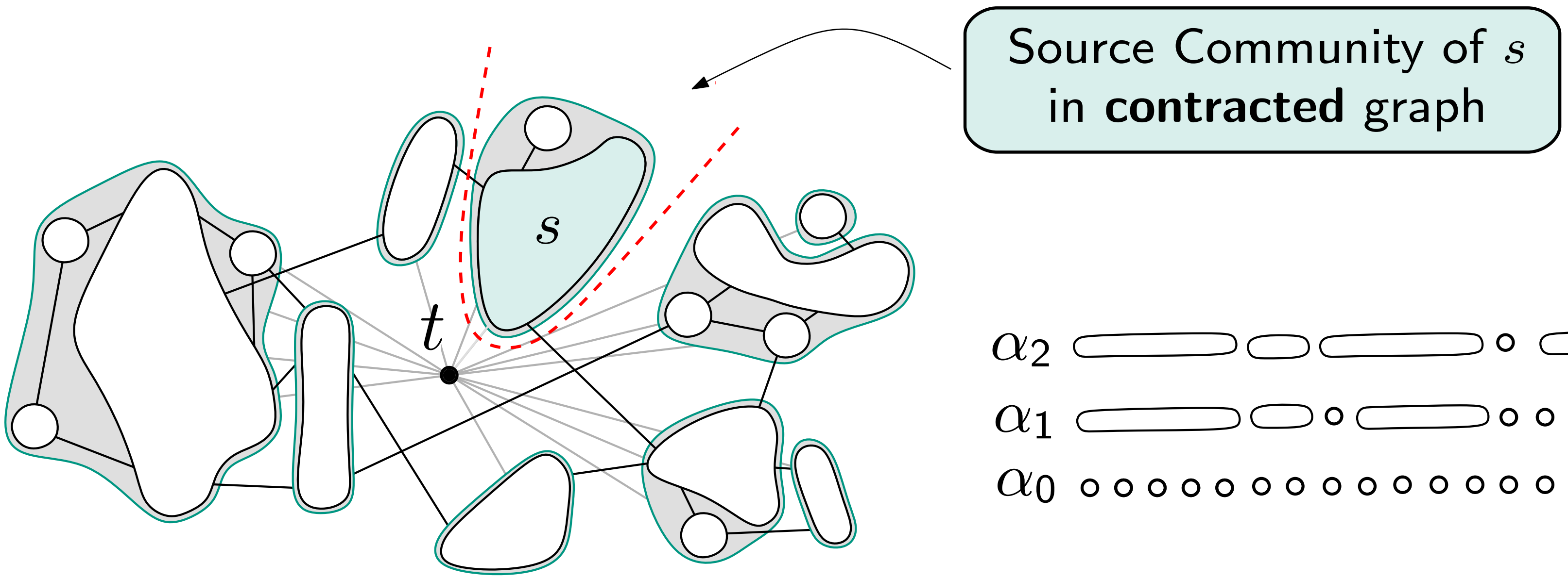


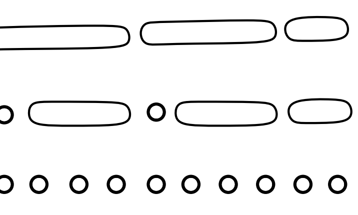
The Cut-Clustering Algorithm

- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

For $i = 1, \ldots, k$ do
- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i

community-cuts are
- unique
- non-crossing





The Cut-Clustering Algorithm

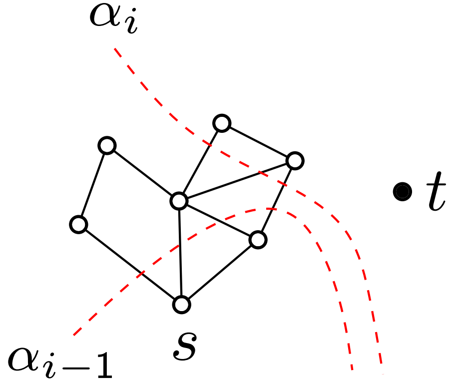
- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

For $i = 1, \ldots, k$ do

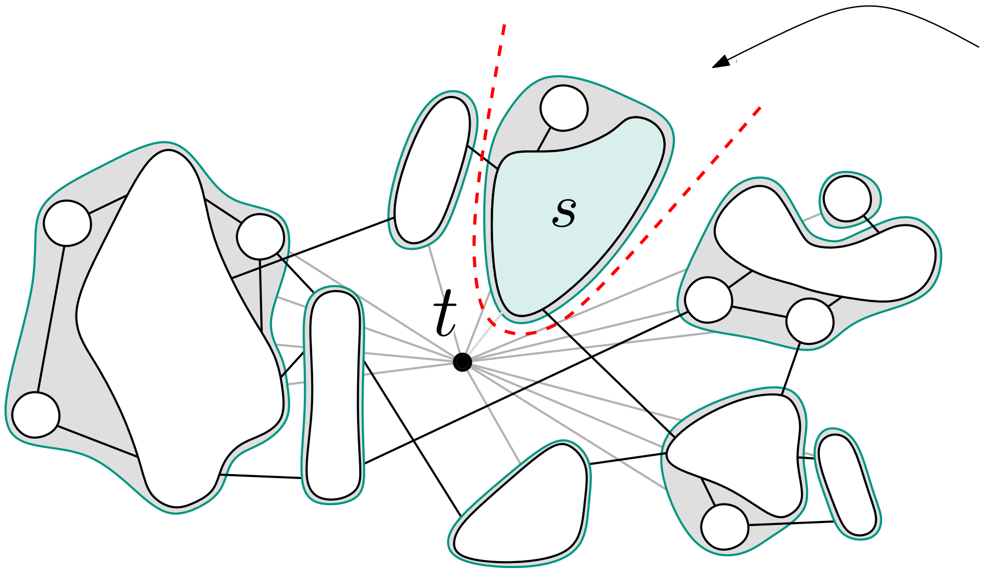
- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i

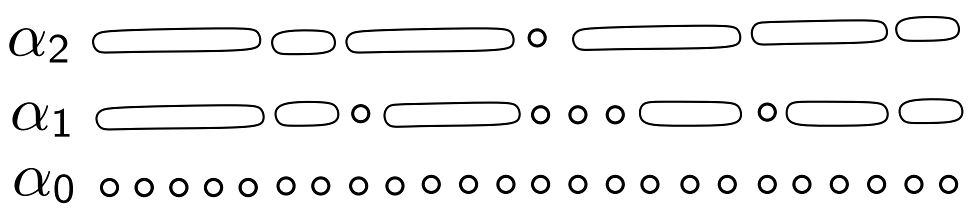
community-cuts are

- unique
- non-crossing
- vertically nested



Source Community of s in **contracted** graph





The Cut-Clustering Algorithm

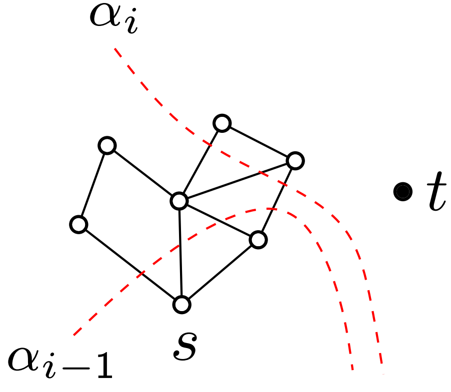
- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

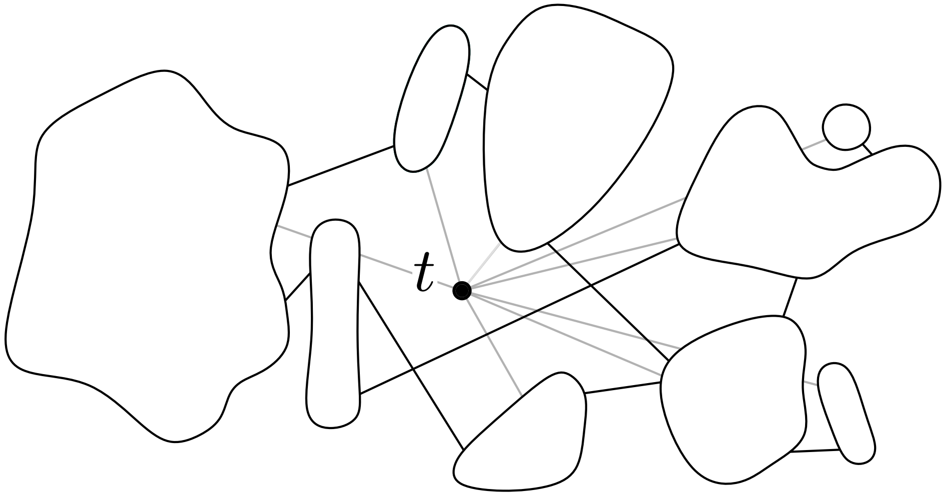
For $i = 1, \ldots, k$ do

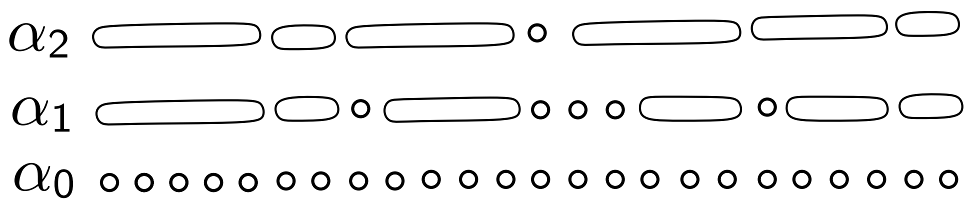
- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i

community-cuts are

- unique
- non-crossing
- vertically nested







The Cut-Clustering Algorithm

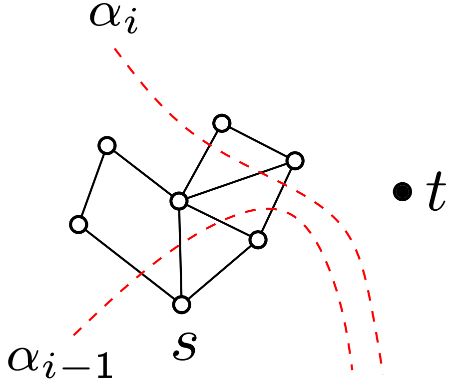
- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

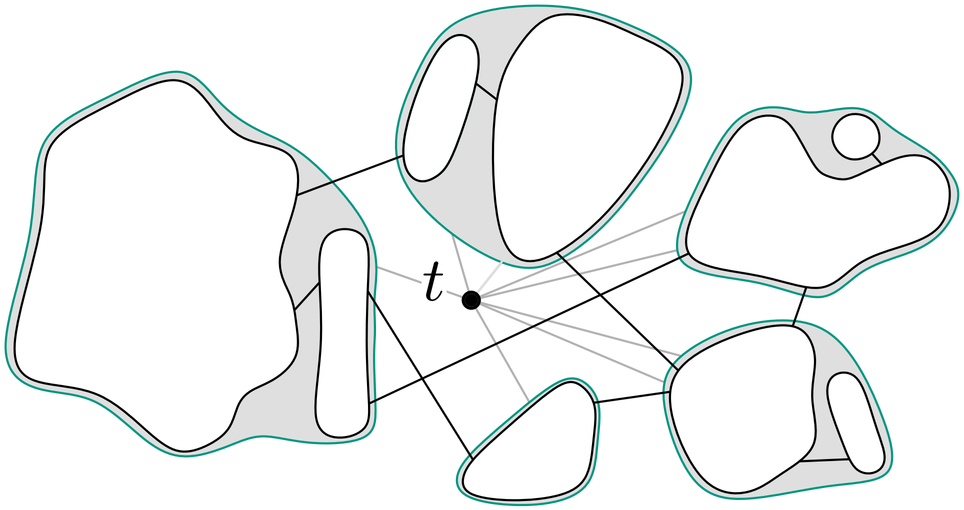
For $i = 1, \ldots, k$ do

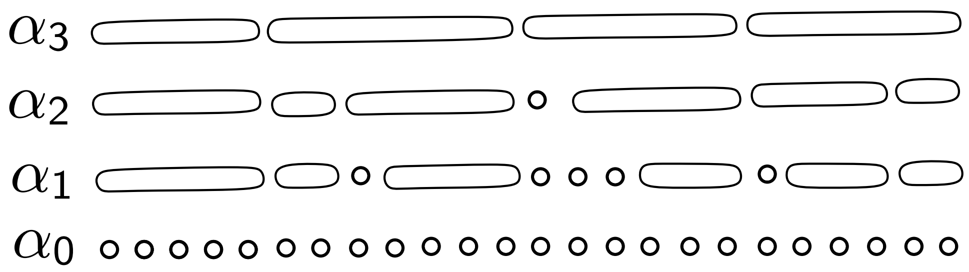
- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i

community-cuts are

- unique
- non-crossing
- vertically nested







The Cut-Clustering Algorithm

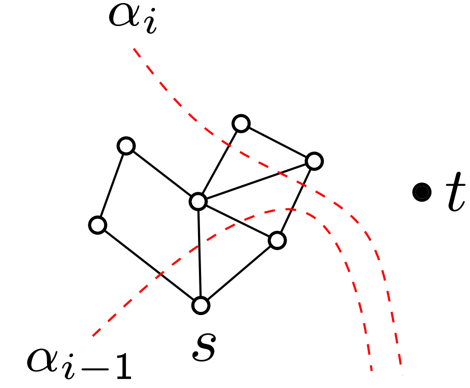
- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

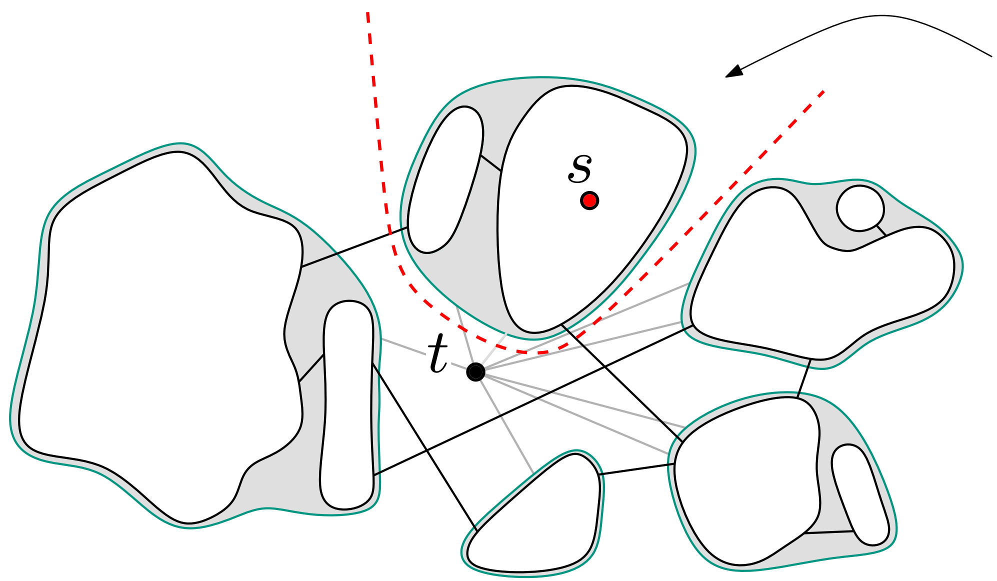
For $i = 1, \ldots, k$ do

- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i

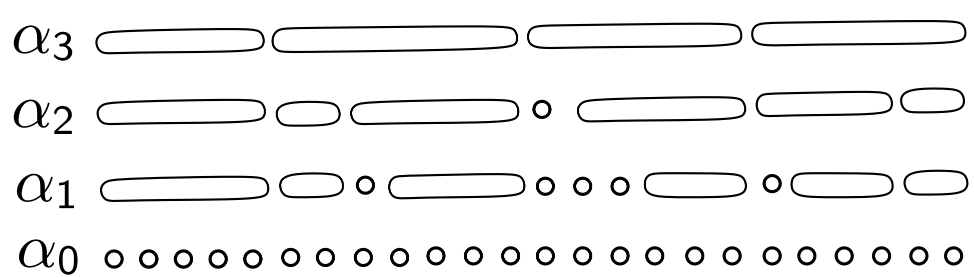
community-cuts are

- unique
- non-crossing
- vertically nested





Source Community of s in **underlying** graph



The Cut-Clustering Algorithm

- Flake et al. [2004]
- iterative application of the Cut-Clustering approach with different values $\alpha_1 > \cdots > \alpha_k$
- $\alpha_0 = 0$, singleton clustering $\mathcal{C}_0$

For $i = 1, \ldots, k$ do

- replace α_{i-1} by α_i
- contract each cluster $C \in \mathcal{C}_{i-1}$
- calculate clustering C_i

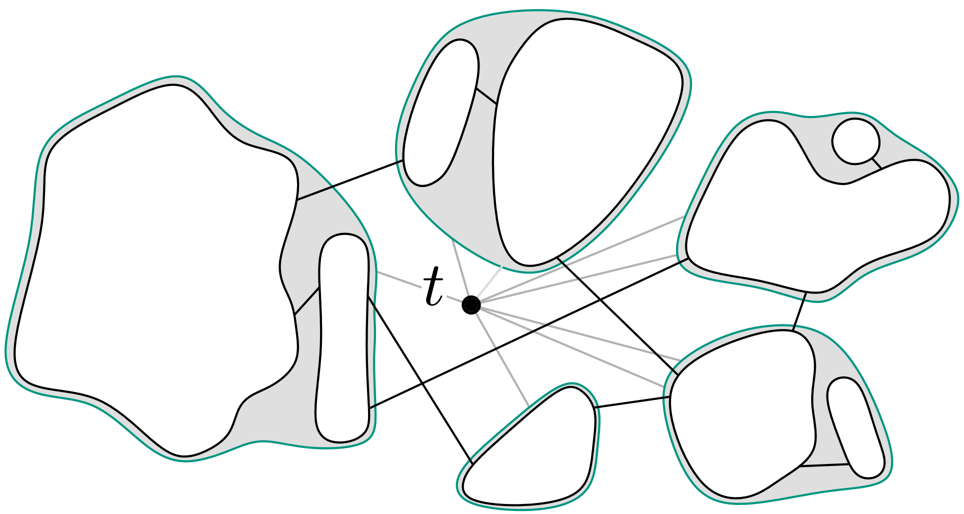
Problem: Choice of α

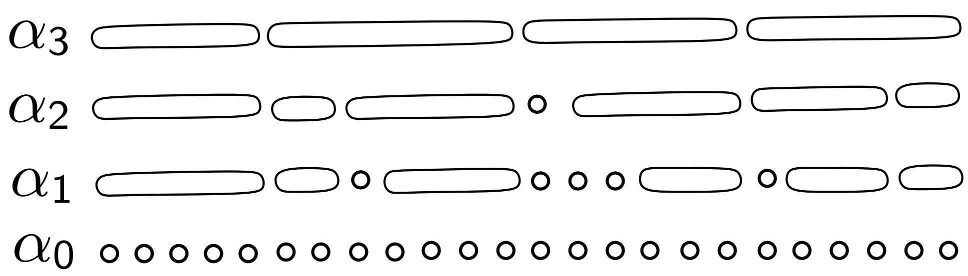
Naive idea: Binary search

needs discretization

too low $\rightarrow$ missing levels

too high $\rightarrow$ extensive running time





New Parametric Search Approach

- bases on cut-cost function

$$\omega_C : \mathbb{R}_0^+ \to [c(C, V \setminus C), \infty) \subset \mathbb{R}_0^+$$

$$\omega_C(\alpha) := c(C, V \setminus C) + |C|\alpha$$

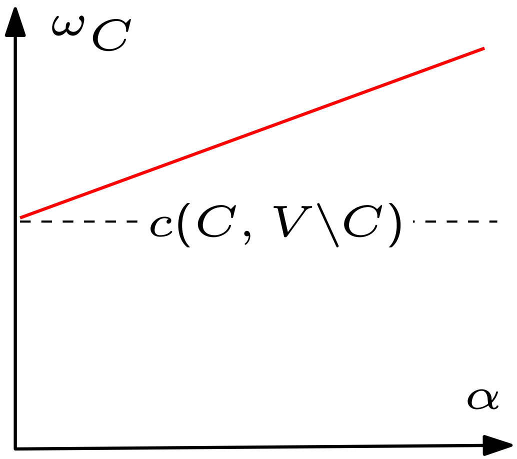
New Parametric Search Approach

- bases on cut-cost function

$$\omega_C : \mathbb{R}_0^+ \to [c(C, V \setminus C), \infty) \subset \mathbb{R}_0^+$$

$$\omega_C(\alpha) := c(C, V \setminus C) + |C|\alpha$$

slope

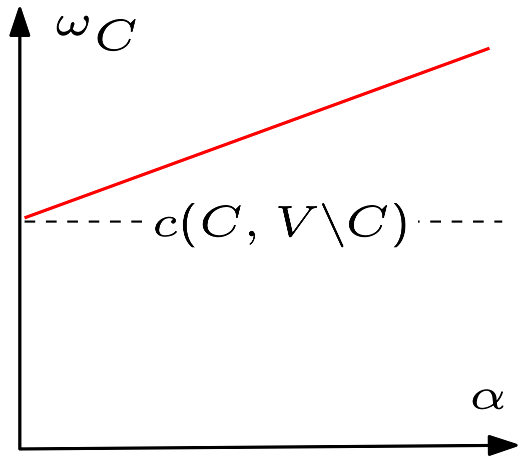


New Parametric Search Approach

- bases on cut-cost function

$$\omega_C : \mathbb{R}_0^+ \to [c(C, V \setminus C), \infty) \subset \mathbb{R}_0^+$$
$$\omega_C(\alpha) := c(C, V \setminus C) + |C|\alpha$$

slope



Theorem:
Let $\mathcal{C}_i < \mathcal{C}_j$ denote two different clusterings with associated parameter values $\alpha_i > \alpha_j$.
In time $O(|\mathcal{C}_i|)$ a parameter value α_m with $\alpha_j < \alpha_m \leq \alpha_i$ can be computed such that

1) $\mathcal{C}_i \leq \mathcal{C}_m < \mathcal{C}_j$ and
2) $\mathcal{C}_m = \mathcal{C}_i$ implies that α_m is the breakpoint between $\mathcal{C}_i$ and $\mathcal{C}_j$.

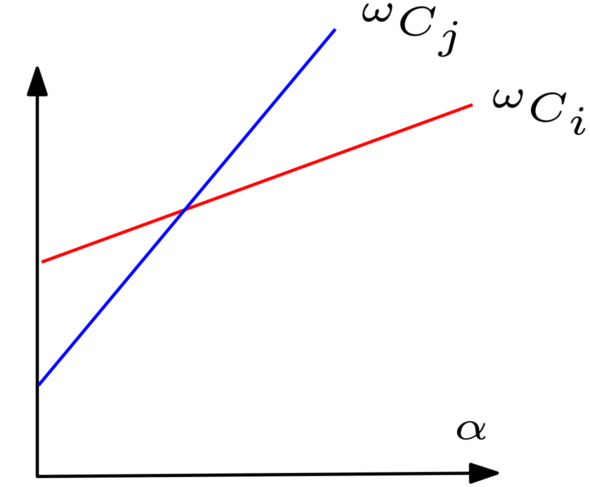
New Parametric Search Approach

- bases on cut-cost function

$$\omega_C : \mathbb{R}_0^+ \to [c(C, V \setminus C), \infty) \subset \mathbb{R}_0^+$$

$$\omega_C(\alpha) := c(C, V \setminus C) + |C|\alpha$$

slope

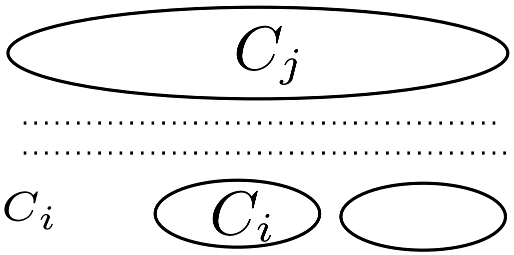


Theorem:
Let $\mathcal{C}_i < \mathcal{C}_j$ denote two different clusterings with associated parameter values $\alpha_i > \alpha_j$.
In time $O(|\mathcal{C}_i|)$ a parameter value α_m with $\alpha_j < \alpha_m \leq \alpha_i$ can be computed such that

1) $\mathcal{C}_i \leq \mathcal{C}_m < \mathcal{C}_j$ and
2) $\mathcal{C}_m = \mathcal{C}_i$ implies that α_m is the breakpoint between $\mathcal{C}_i$ and $\mathcal{C}_j$.

$$\alpha_m := \min_{C_j \in \mathcal{C}_j} \{\max_{C_i \subset C_j} \alpha_i'\}$$

with α_i' the intersection point of ω_{C_j} and ω_{C_i}



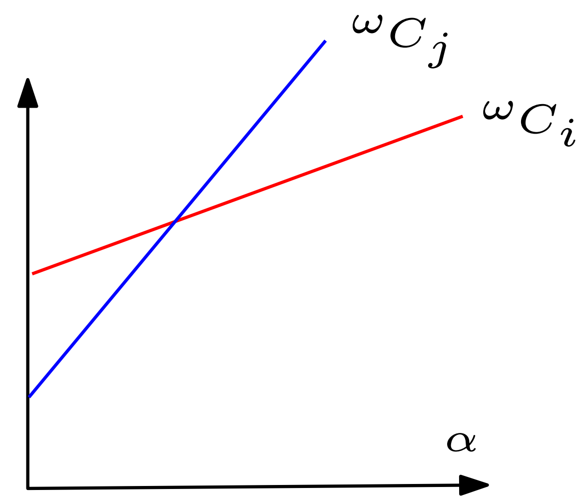
New Parametric Search Approach

- bases on cut-cost function

$$\omega_C : \mathbb{R}_0^+ \to [c(C, V \setminus C), \infty) \subset \mathbb{R}_0^+$$

$$\omega_C(\alpha) := c(C, V \setminus C) + |C|\alpha$$

slope

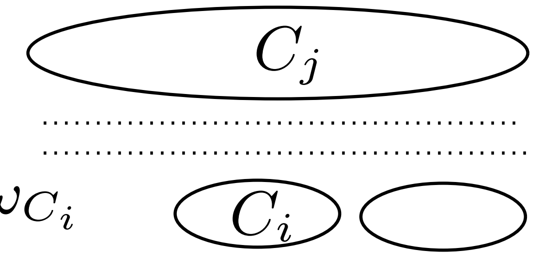


Theorem:
Let $\mathcal{C}_i < \mathcal{C}_j$ denote two different clusterings with associated parameter values $\alpha_i > \alpha_j$.
In time $O(|\mathcal{C}_i|)$ a parameter value α_m with $\alpha_j < \alpha_m \leq \alpha_i$ can be computed such that

1) $\mathcal{C}_i \leq \mathcal{C}_m < \mathcal{C}_j$ and
2) $\mathcal{C}_m = \mathcal{C}_i$ implies that α_m is the breakpoint between $\mathcal{C}_i$ and $\mathcal{C}_j$.

$$\alpha_m := \min_{C_j \in \mathcal{C}_j} \{\max_{C_i \subset C_j} \alpha_i'\}$$

with α_i' the intersection point of ω_{C_j} and ω_{C_i}



$\Rightarrow$ at most $2h$ iterations, returns **complete** hierarchy

Running Time Experiment

graph	n	m	h	ParS [m:s]	BinS [fac]
jazz	198	2742	3	0.062	7.871
celegans_metabolic	453	2025	8	0.300	8.380
celegansneural	297	2148	17	0.406	9.919
email	1133	5451	4	1.116	8.758
netscience	1589	2742	38	4.310	11.952
bo_cluster	2114	2277	19	4.355	12.800
data	2851	15093	4	11.506	9.620
dokuwiki_org	4416	12914	18	39.815	15.571
power	4941	6594	66	1:25.736	15.742
hep-th	8361	15751	56	6:26.213	18.183
PGPgiantcompo	10680	24316	94	13:25.121	18.575
as-22july06	22963	48436	33	39:54.495	20.583
cond-mat	16726	47594	80	44:15.317	27.425
astro-ph	16706	121251	60	98:25.791	24.825
rgg_n_2_15	32768	160240	46	245:25.644	22.573
cond-mat-2003	31163	120029	74	268:14.601	20.933
G_n_pin_pout	100000	501198	4	369:29.033	*
cond-mat-2005	40421	175691	82	652:32.163	21.446

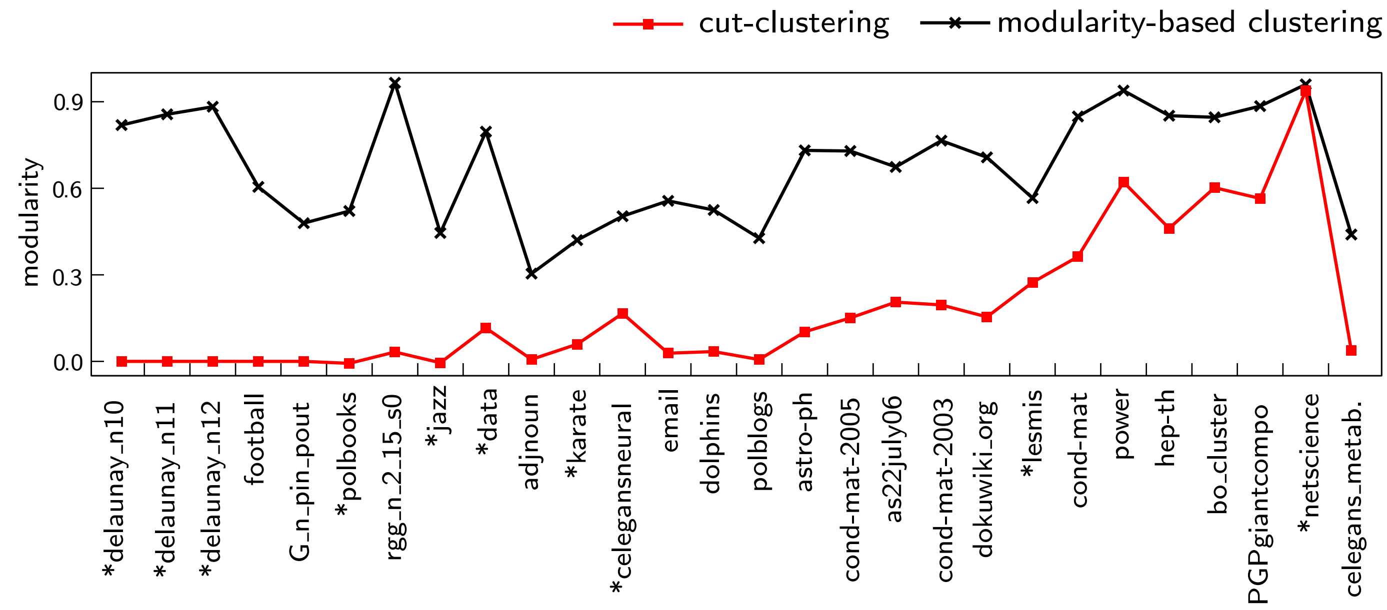
Running Time Experiment

graph	n	m	h	ParS [m:s]	BinS [fac]
jazz	198	2742	3	0.062	7.871
celegans_metabolic	453	2025	8	0.300	8.380
celegansneural	297	2148	17	0.406	9.919
email	1133	5451	4	1.116	8.758
netscience	1589	2742	38	4.310	11.952
bo_cluster	2114	2277	19	4.355	12.800
data	2851	15093	4	11.506	9.620
dokuwiki_org	4416	12914	18	39.815	15.571
power	4941	6594	66	1:25.736	15.742
hep-th	8361	15751	56	6:26.213	18.183
PGPgiantcompo	10680	24316	94	13:25.121	18.575
as-22july06	22963	48436	33	39:54.495	20.583
cond-mat	16726	47594	80	44:15.317	27.425
astro-ph	16706	121251	60	98:25.791	24.825
rgg_n_2_15	32768	160240	46	245:25.644	22.573
cond-mat-2003	31163	120029	74	268:14.601	20.933
G_n_pin_pout	100000	501198	4	369:29.033	*
cond-mat-2005	40421	175691	82	652:32.163	21.446

Modularity Analysis

cut-clustering algorithm vs. modularity-based greedy approach

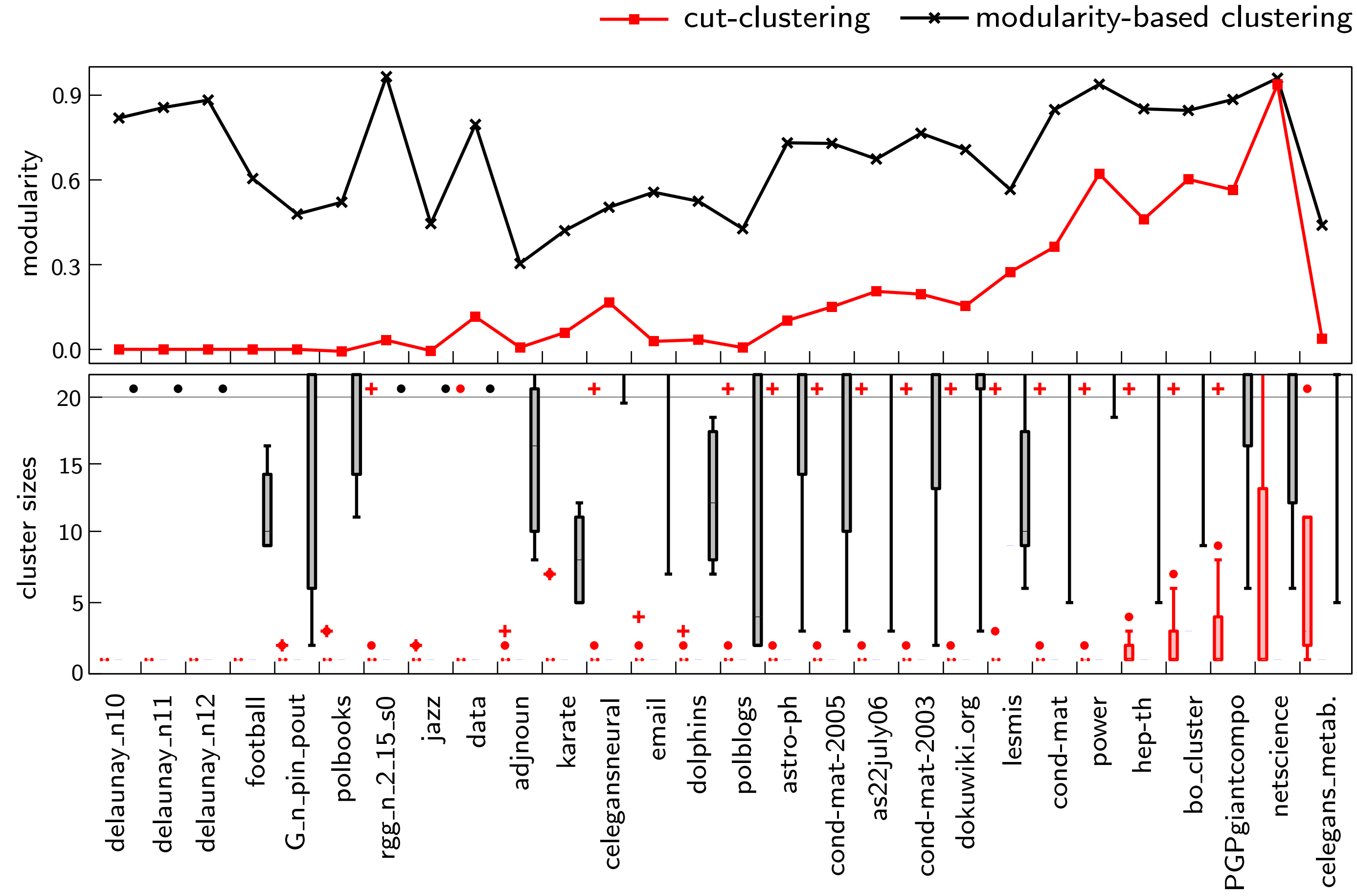
- modClus $\geq$ cutClus



Modularity Analysis

cut-clustering algorithm vs. modularity-based greedy approach

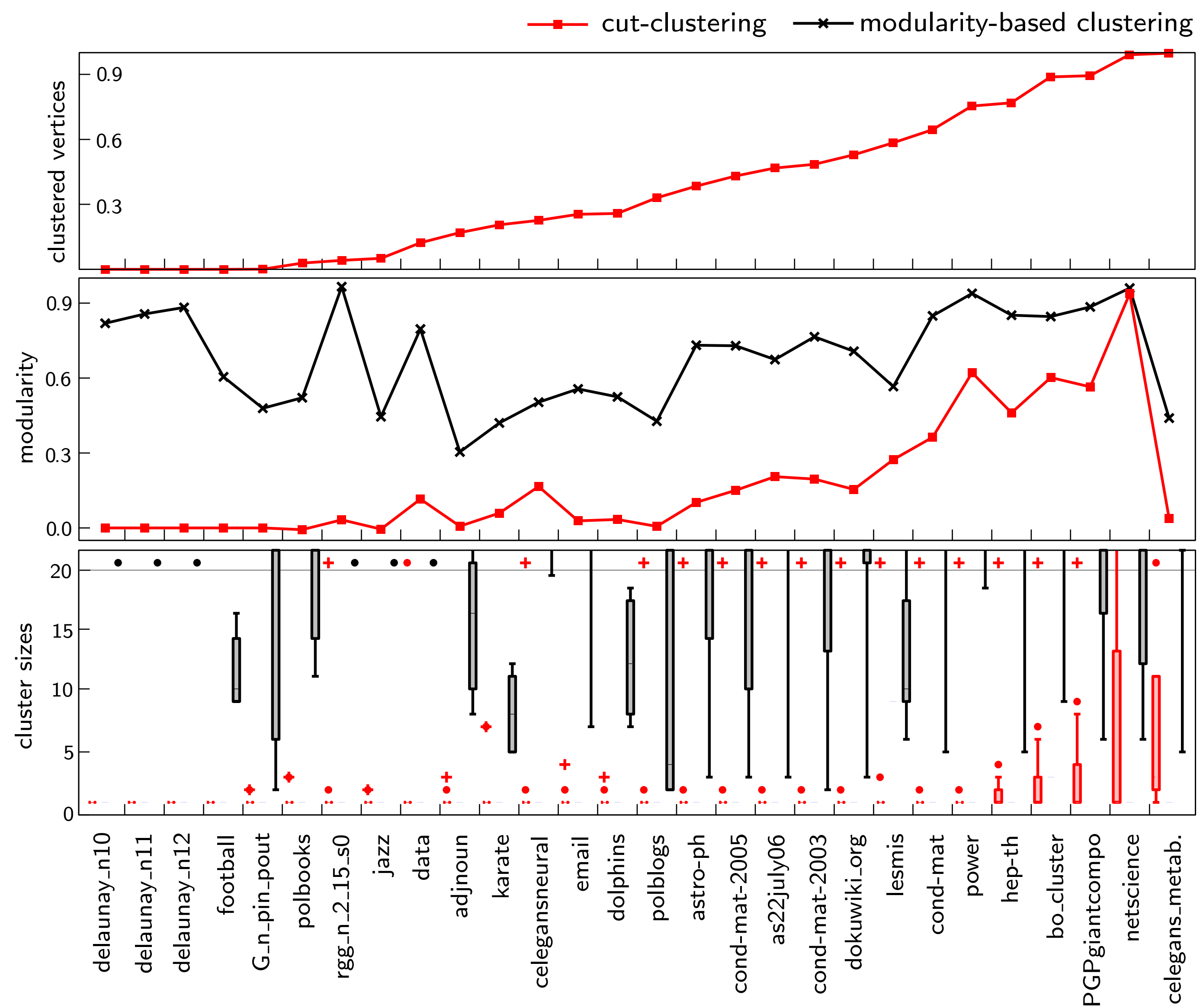
- modClus $\geq$ cutClus
- cutClus increases with coarseness



Modularity Analysis

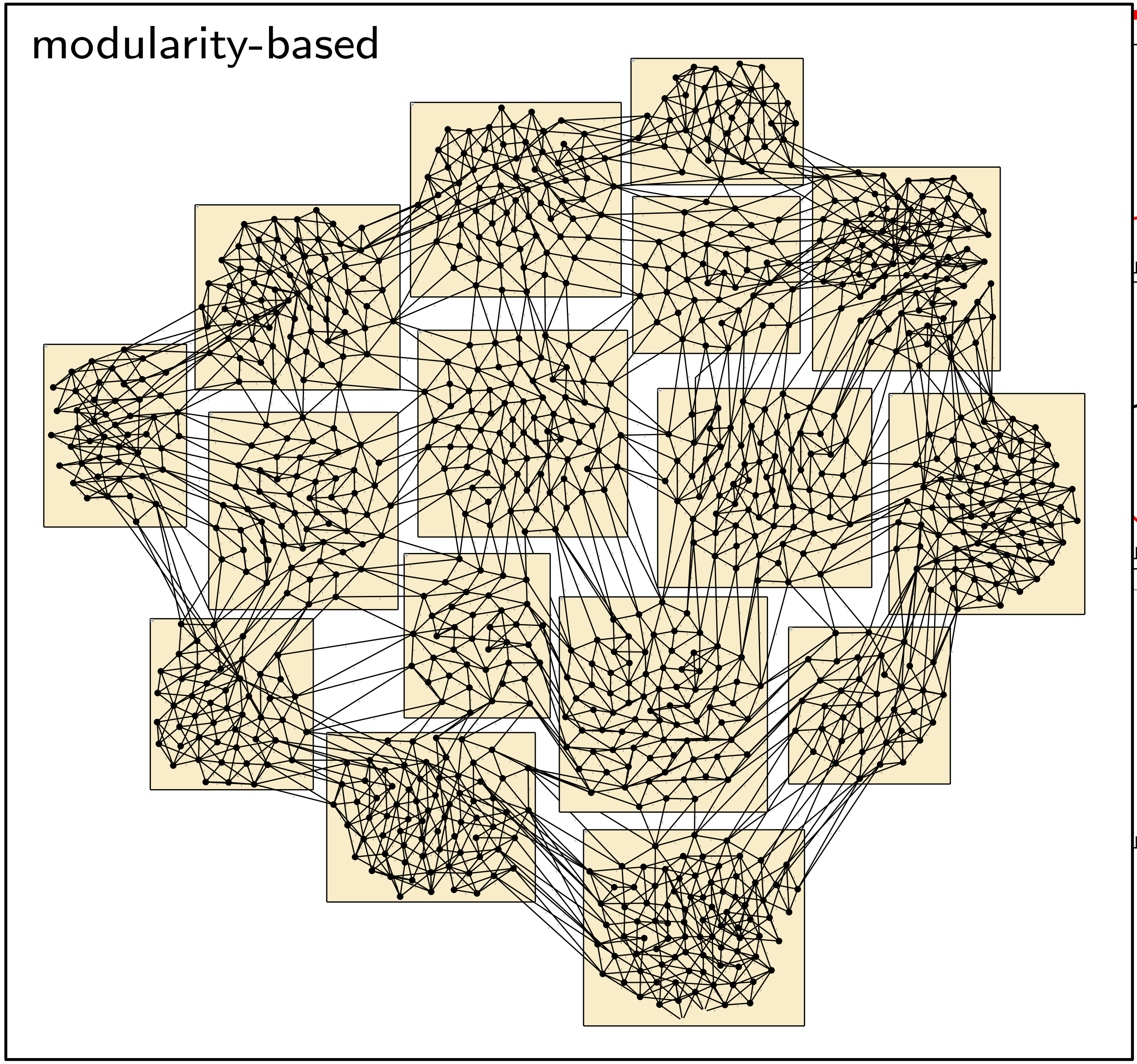
cut-clustering algorithm vs. modularity-based greedy approach

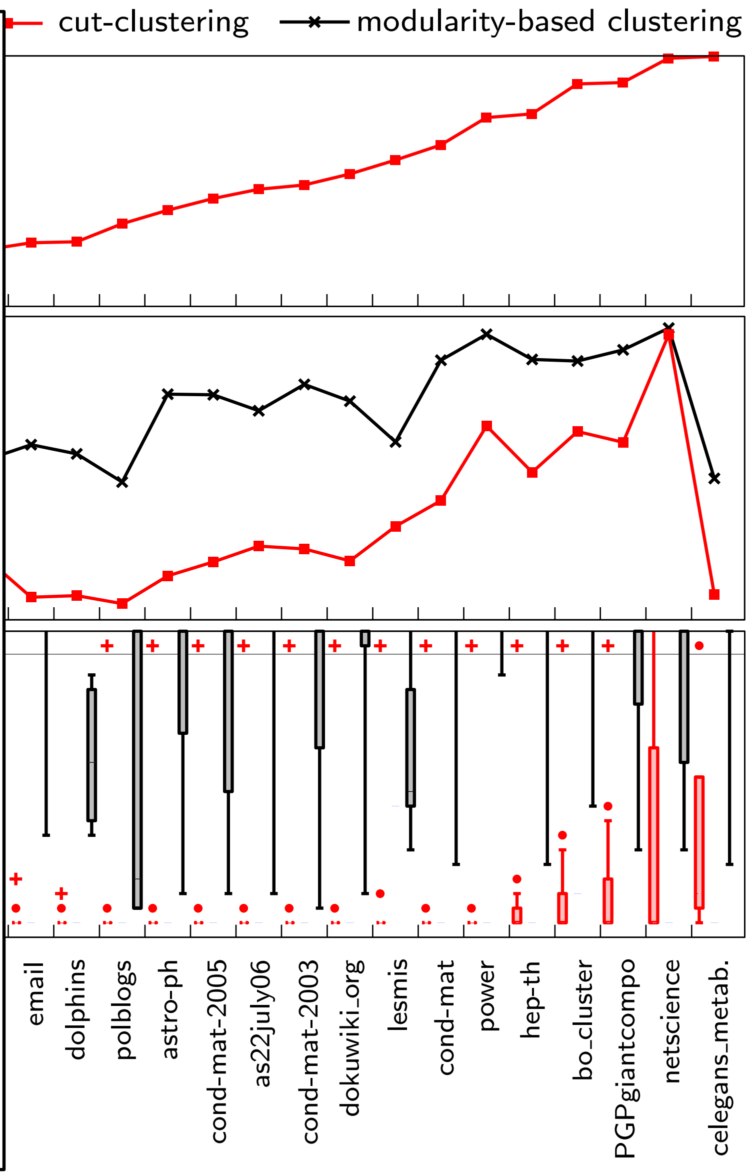
- modClus $\geq$ cutClus
- cutClus increases with coarseness



Modularity Analysis

cut-clustering algorithm vs. modularity-based greedy approach

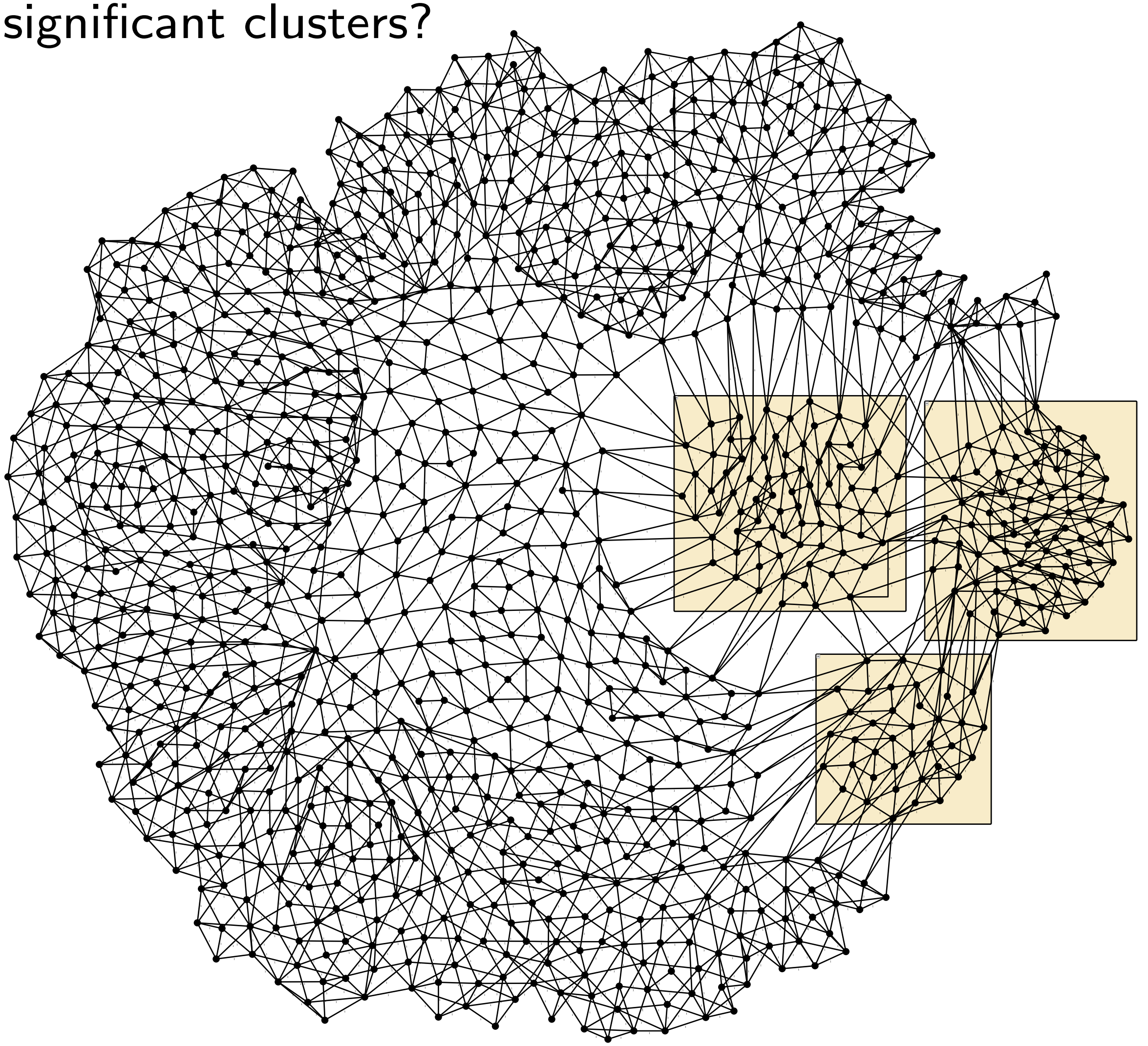


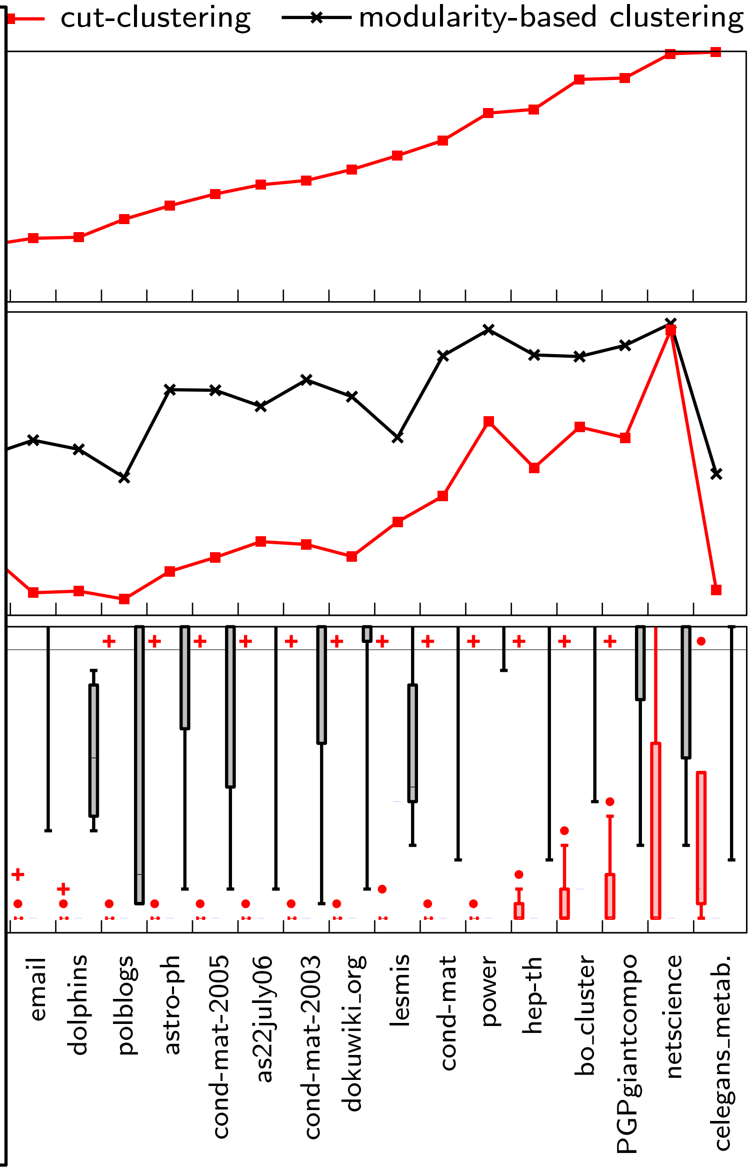


Modularity Analysis

cut-clustering algorithm vs. modularity-based greedy approach

significant clusters?

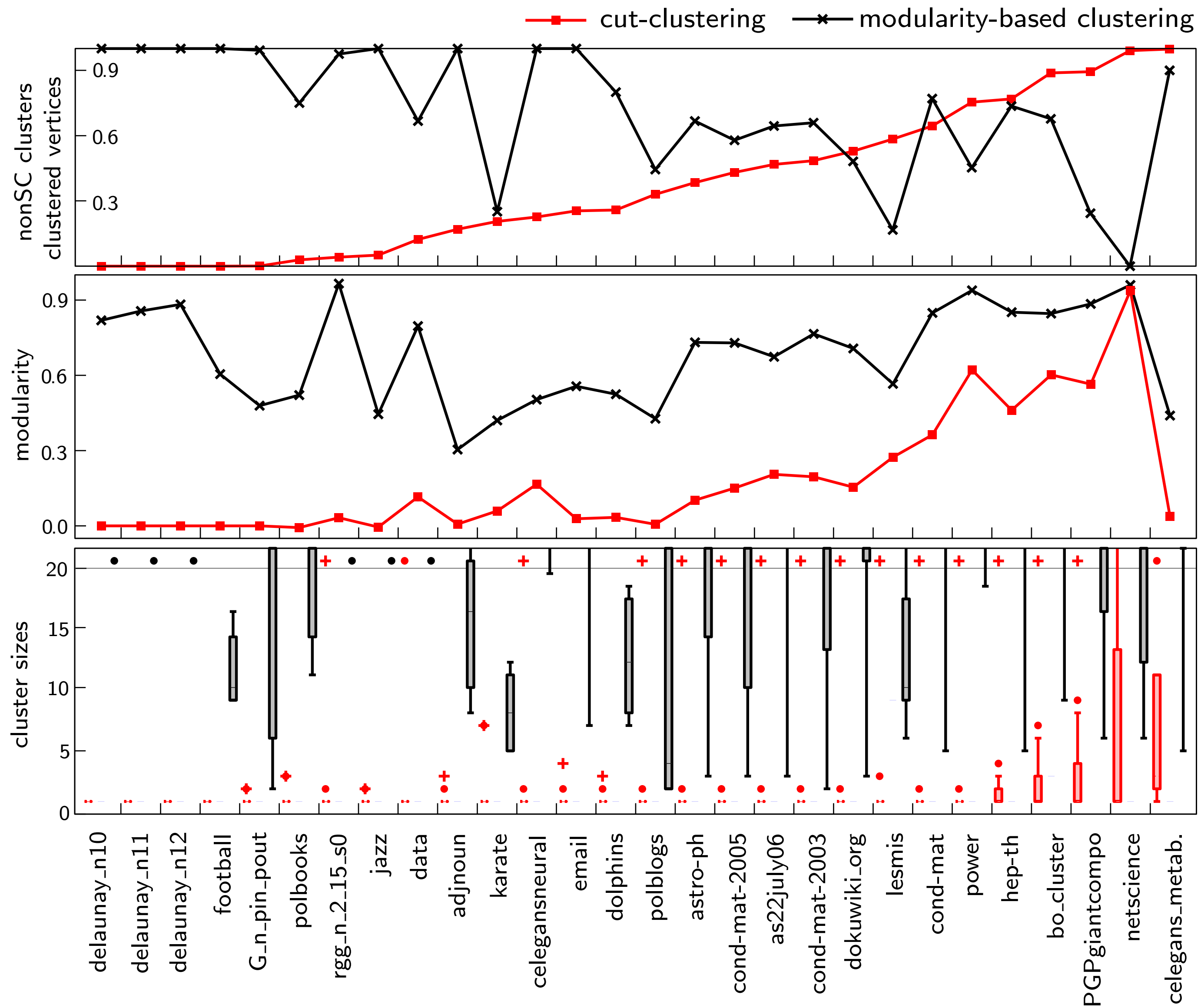




Modularity Analysis

Source-community S of s:
$\forall\, U \subset S \backslash \{s\}$: $c(U, S\backslash U) \geq c(U, V\backslash S)$

- modClus $\geq$ cutClus
- cutClus increases with coarseness
- modClus often misses source-community property

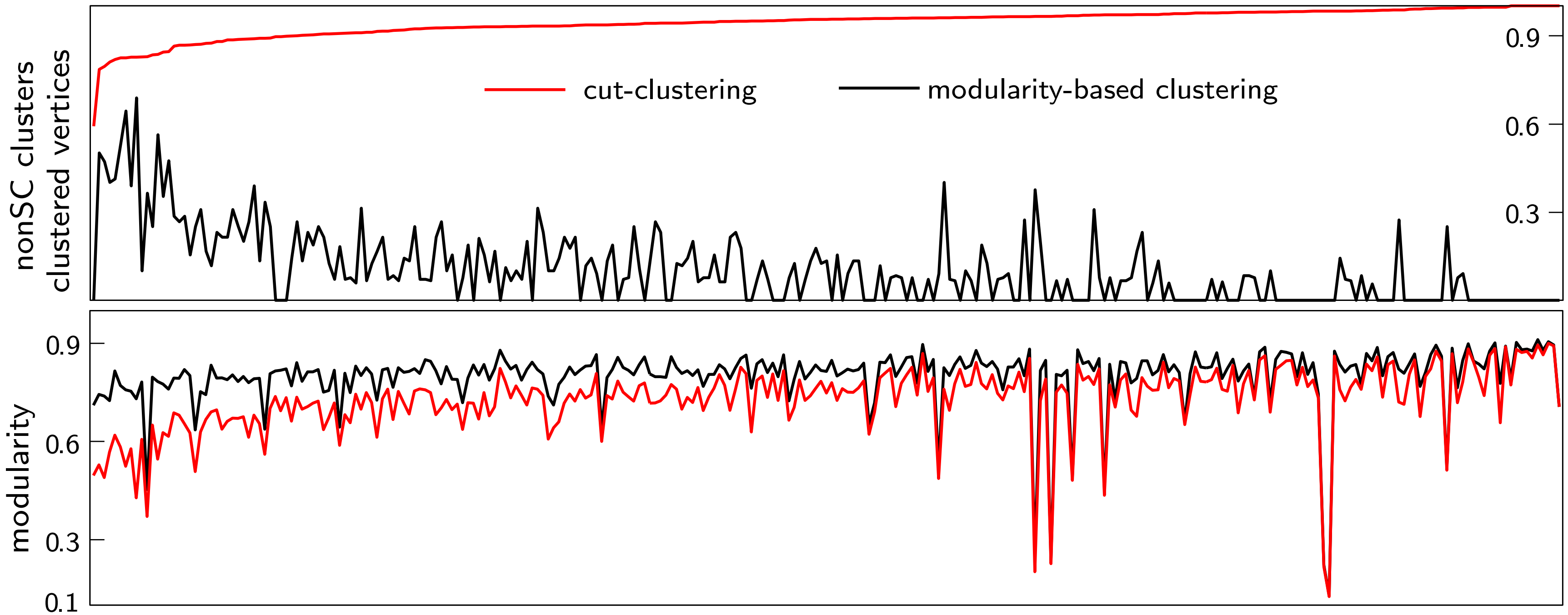


Modularity Analysis

Source-community S of s:
$\forall\, U \subset S \backslash \{s\}$: $c(U, S\backslash U) \geq c(U, V\backslash S)$

- modClus $\geq$ cutClus
- cutClus increases with coarseness
- modClus often misses source-community property

275 snapshots of the email network of the Department of Informarics at KIT.

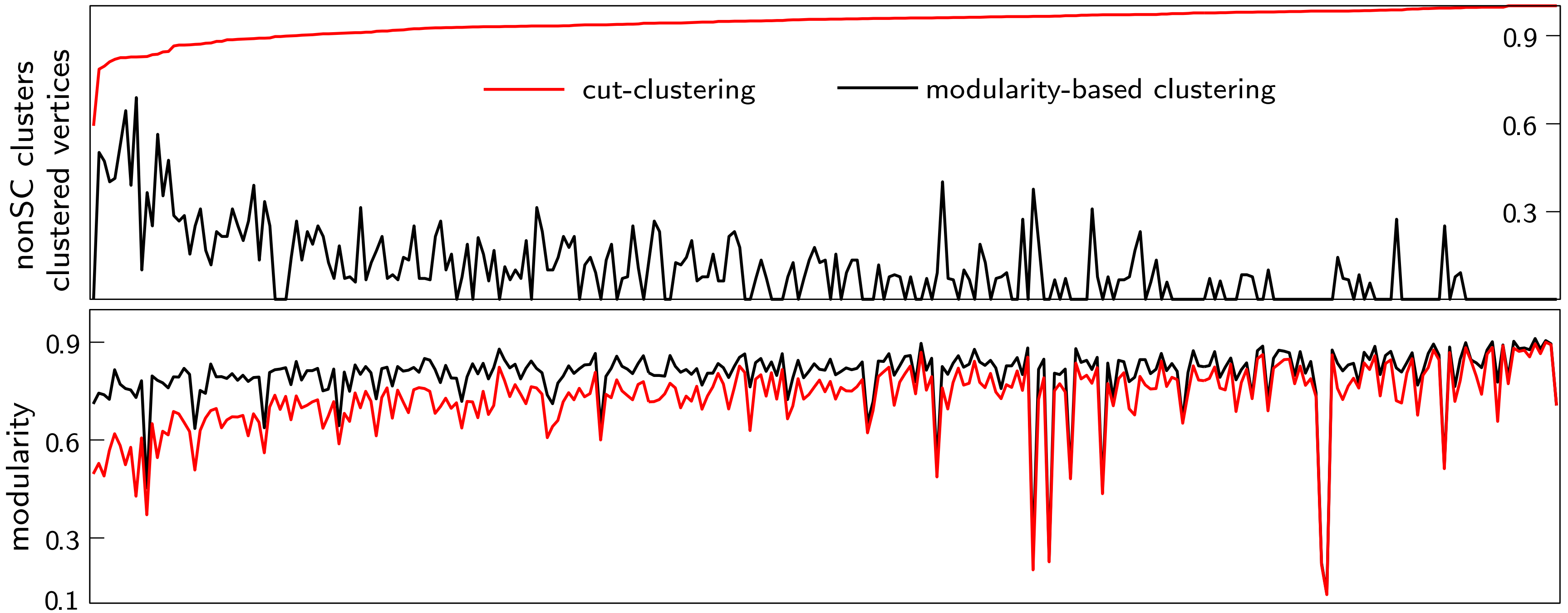


Modularity Analysis

Source-community S of s:
$\forall\, U \subset S \setminus \{s\}\colon\ c(U, S\setminus U) \geq c(U, V\setminus S)$

- modClus $\geq$ cutClus
- cutClus increases with coarseness
- modClus often misses source-community property
- same trend for both
- cutClus close to modClus
- modClus closer to source-community property

275 snapshots of the email network of the Department of Informarics at KIT.



Modularity Analysis

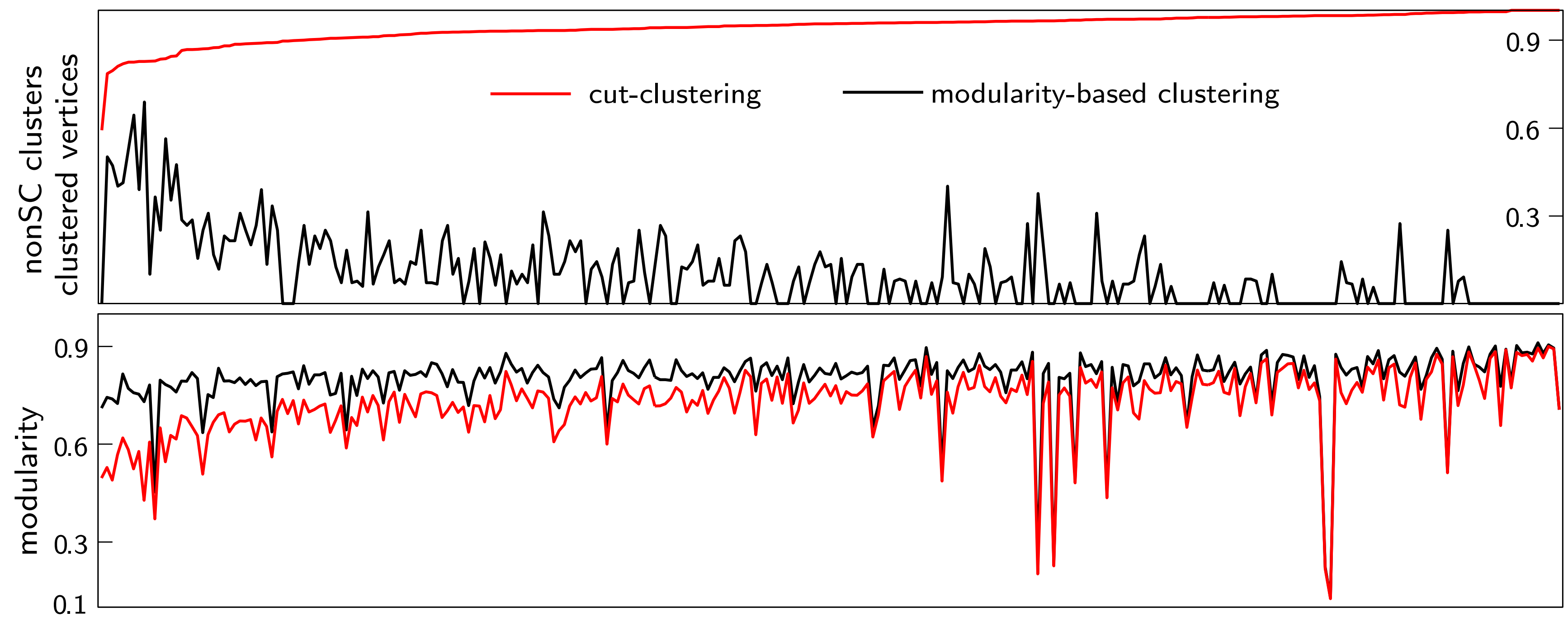
Source-community S of s:
$\forall\, U \subset S \setminus \{s\}$: $c(U, S \setminus U) \geq c(U, V \setminus S)$

- modClus $\geq$ cutClus
- cutClus increases with coarseness
- modClus often misses source-community property

- same trend for both
- cutClus close to modClus
- modClus closer to source-community property

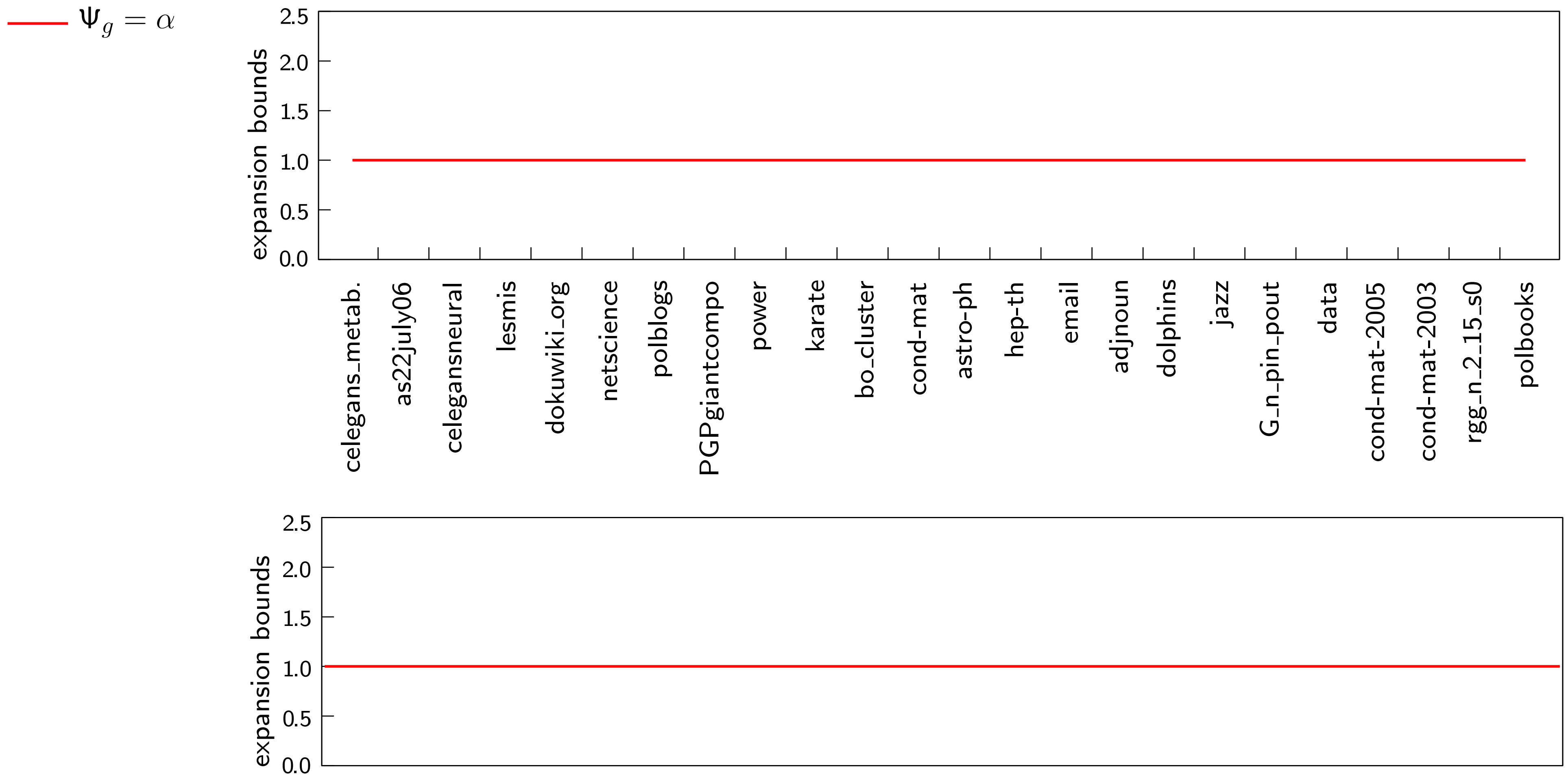
- high dependence on graph structure
- Conjecture: Good cutClus if significant clustering exists

275 snapshots of the email network of the Department of Informarics at KIT.



Expansion Analysis

intra-cluster expansion is
NP-hard to compute

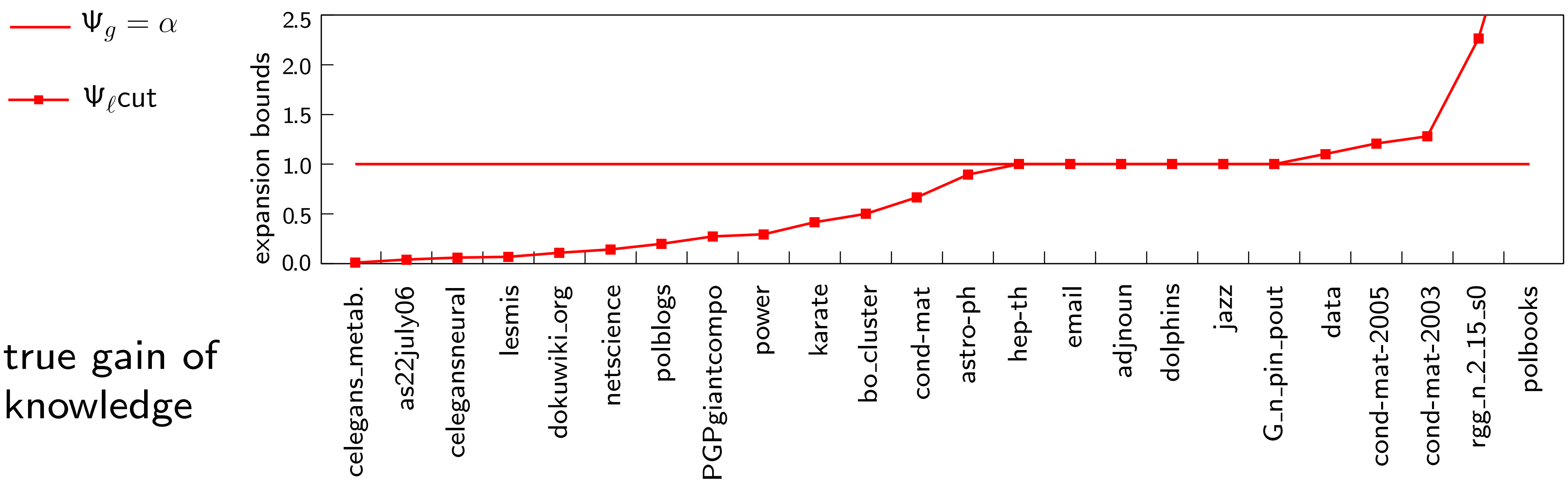


275 snapshots of the email network of the Department of Informarics at KIT.

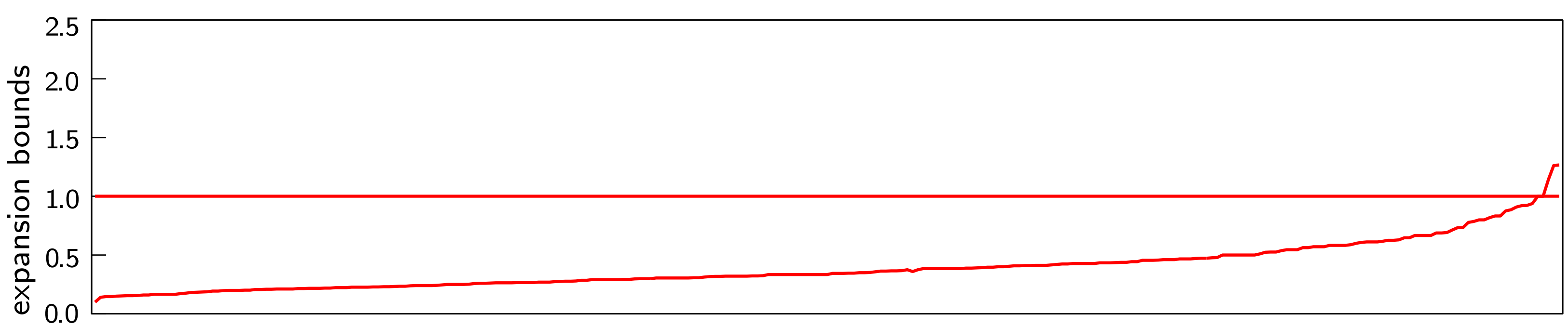
Expansion Analysis

intra-cluster expansion is
NP-hard to compute

trivial lower bound: $\Psi_\ell(\mathcal{C}) := \min_{C \in \mathcal{C}} \{ \frac{c(S, C \setminus S)}{|C|/2} \}$ with S global min-cut in C



- true gain of knowledge



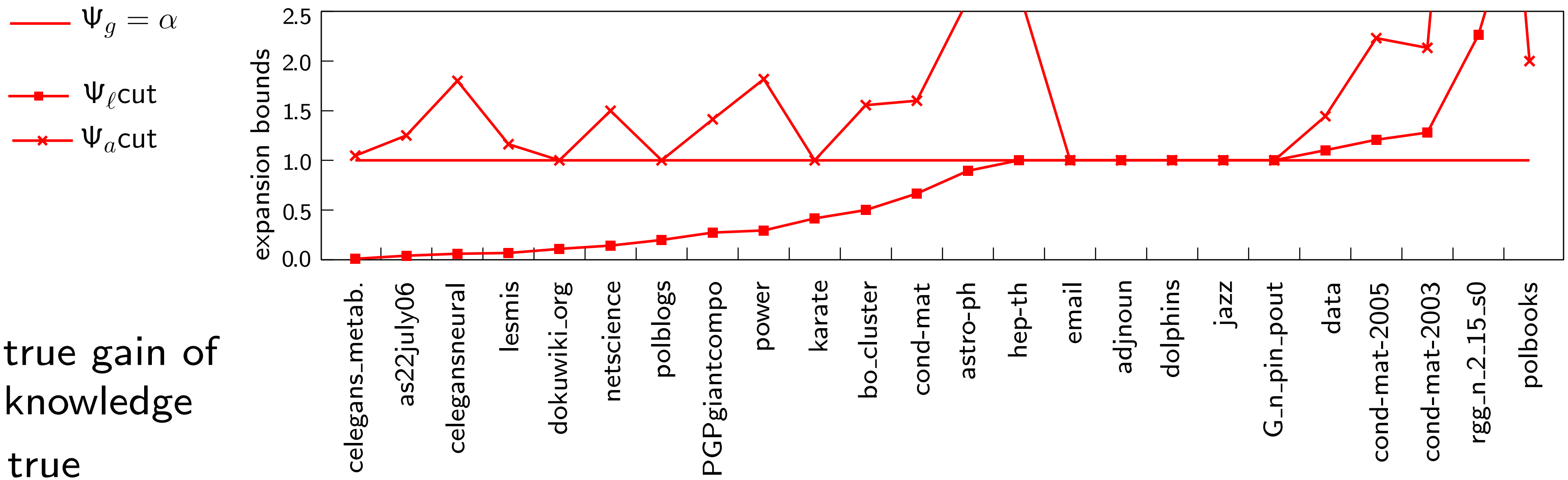
275 snapshots of the email network of the Department of Informarics at KIT.

Expansion Analysis

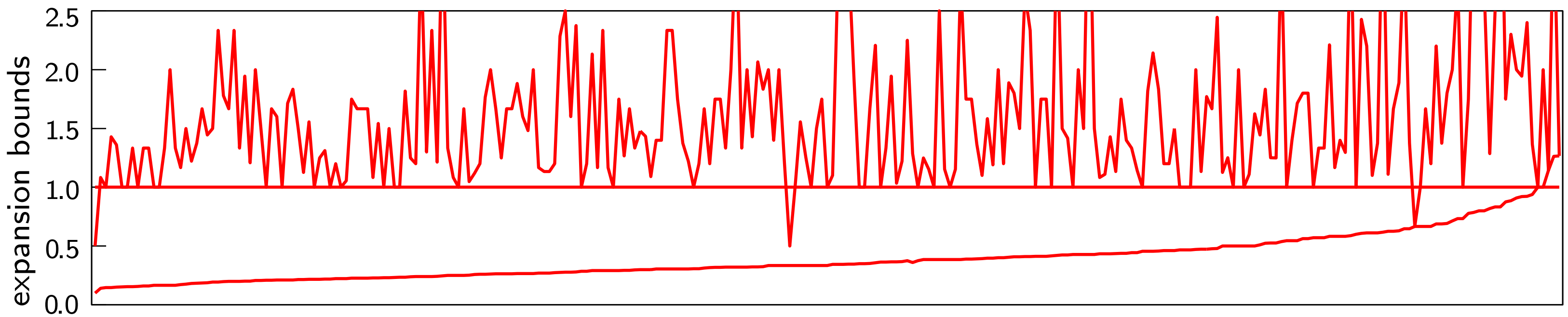
intra-cluster expansion is NP-hard to compute

trivial lower bound: $\Psi_\ell(\mathcal{C}) := \min_{C\in\mathcal{C}}\{\frac{c(S,C\setminus S)}{|C|/2}\}$ with S global min-cut in C

nontrivial lower bound Ψ_a: individually applying cutClus to Clusters



- true gain of knowledge
- true expansion even better



275 snapshots of the email network of the Department of Informarics at KIT.

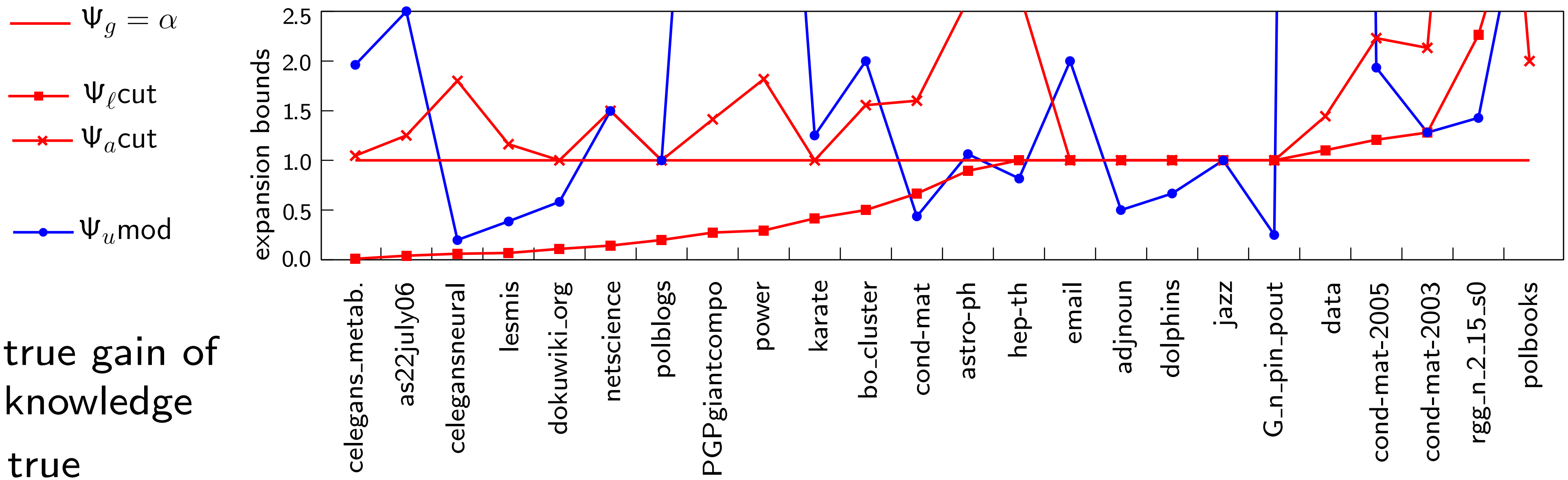
Expansion Analysis

intra-cluster expansion is NP-hard to compute

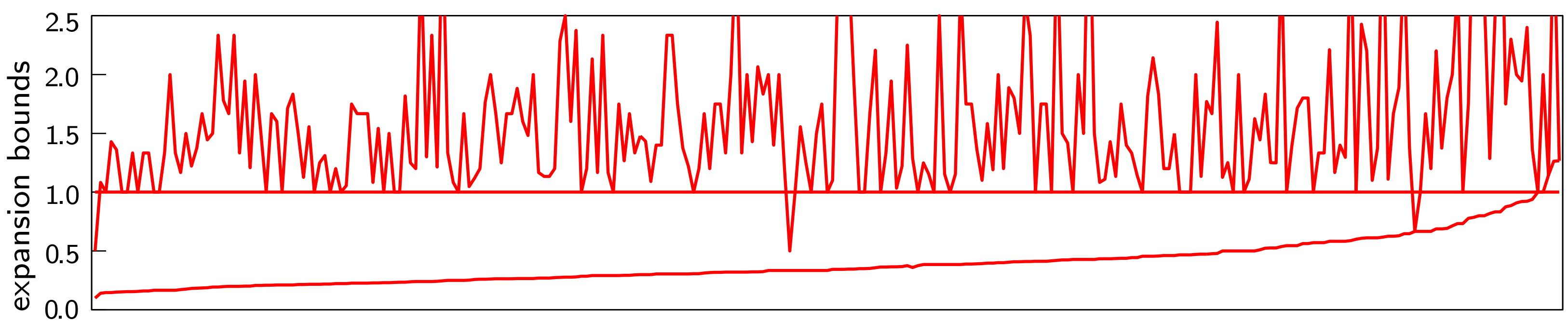
trivial lower bound: $\Psi_\ell(\mathcal{C}) := \min_{C\in\mathcal{C}}\{\frac{c(S,C\setminus S)}{|C|/2}\}$ with S global min-cut in C

nontrivial lower bound Ψ_a: individually applying cutClus to Clusters

trivial upper bound: $\Psi_u(\mathcal{C}) := \min_{C\in\mathcal{C}}\{\frac{c(S,C\setminus S)}{\min\{|S|,|C\setminus S|\}}\}$ with S global min-cut in C



- true gain of knowledge
- true expansion even better



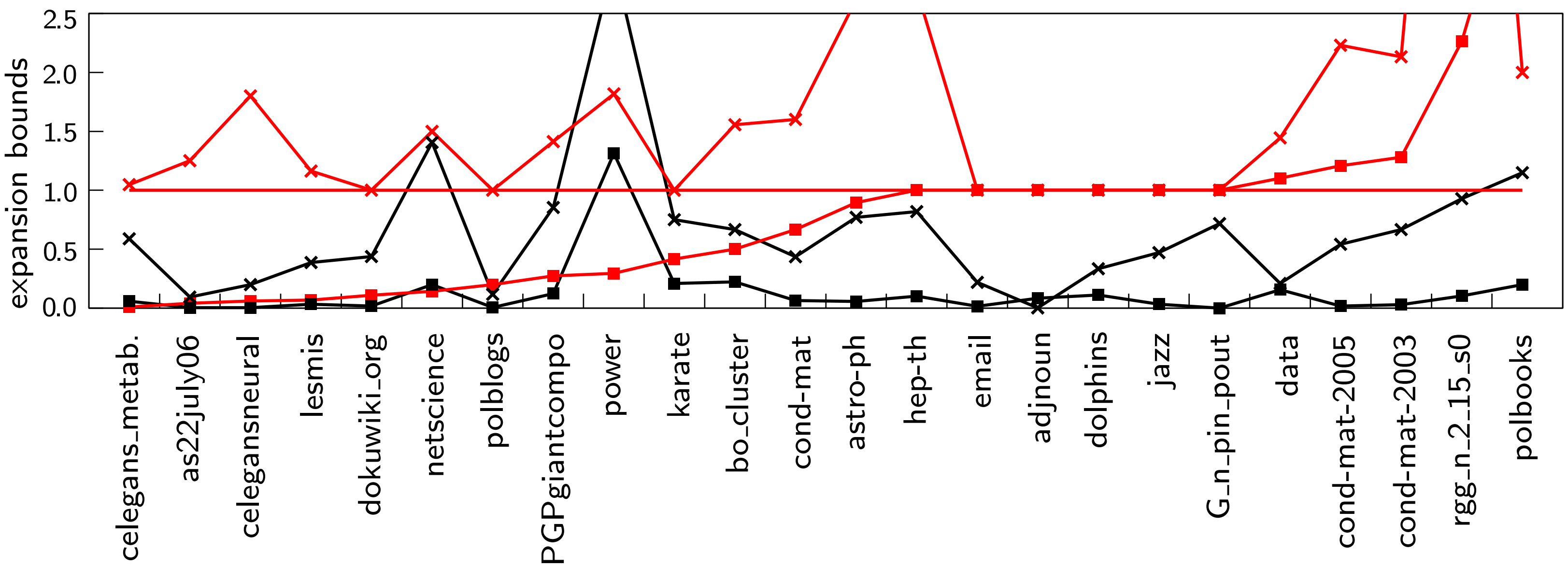
275 snapshots of the email network of the Department of Informarics at KIT.

Expansion Analysis

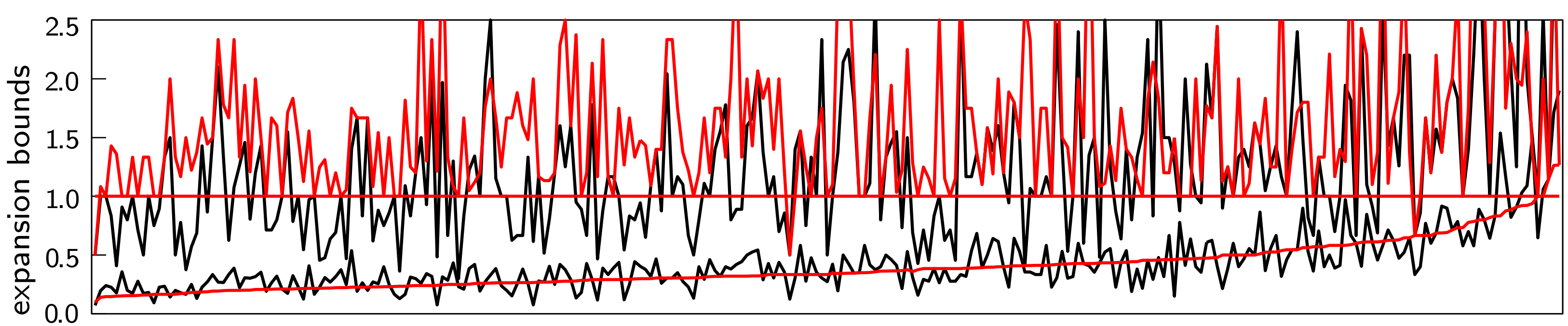
intra-cluster expansion is NP-hard to compute

trivial lower bound: $\Psi_\ell(\mathcal{C}) := \min_{C\in\mathcal{C}}\{\frac{c(S,C\setminus S)}{|C|/2}\}$ with S global min-cut in C

nontrivial lower bound Ψ_a: individually applying cutClus to Clusters



- true gain of knowledge
- true expansion even better
- cutClus $\geq$ modClus



275 snapshots of the email network of the Department of Informarics at KIT.

Conclusion

Analysis of the Hierarchical Cut-Clustering Algorithm of Flake et al. [2004]

- Efficient Parametric Search Approach
  $\rightarrow$ complete hierarchy
- Comparison to modularity-based greedy approach
  $\rightarrow$ competes well if graph structure allows
- Study of guarantee on intra-cluster expansion
  $\rightarrow$ true gain of knowledge

Conclusion

Analysis of the Hierarchical Cut-Clustering Algorithm of Flake et al. [2004]

- Efficient Parametric Search Approach
  $\rightarrow$ complete hierarchy
- Comparison to modularity-based greedy approach
  $\rightarrow$ competes well if graph structure allows
- Study of guarantee on intra-cluster expansion
  $\rightarrow$ true gain of knowledge

Hierarchical Cut-Clustering Algorithm fairly well combines

- **source-communities**, which ensure a good **significance**, with
- decent **modularity** values and
- a guaranteed **intra-cluster expansion**.

Conclusion

Analysis of the Hierarchical Cut-Clustering Algorithm of Flake et al. [2004]

- Efficient Parametric Search Approach
  $\rightarrow$ complete hierarchy
- Comparison to modularity-based greedy approach
  $\rightarrow$ competes well if graph structure allows
- Study of guarantee on intra-cluster expansion
  $\rightarrow$ true gain of knowledge

Hierarchical Cut-Clustering Algorithm fairly well combines

- **source-communities**, which ensure a good **significance**, with
- decent **modularity** values and
- a guaranteed **intra-cluster expansion**.

Thank You