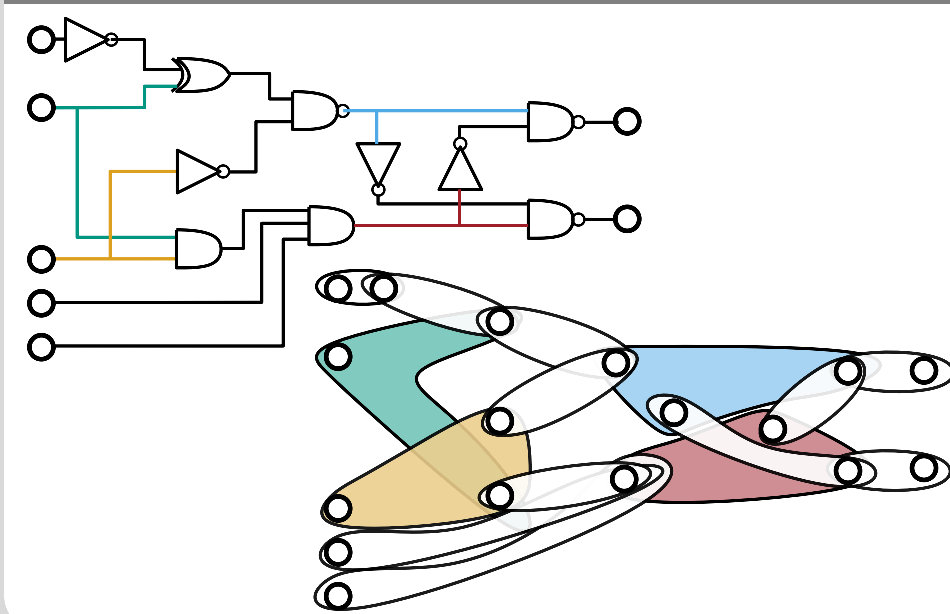
Karlsruhe Institute of Technology

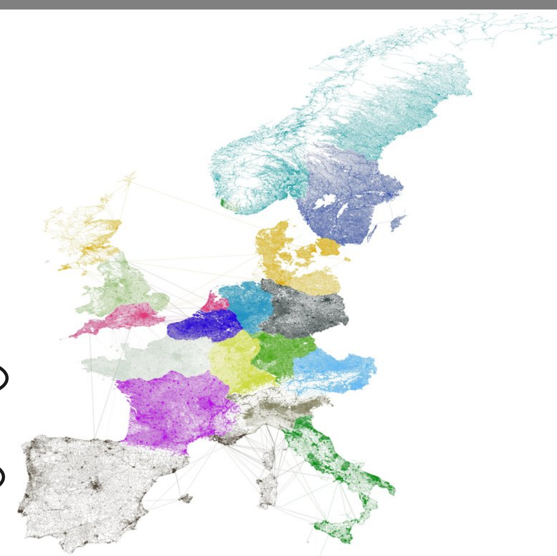
Recent Advances in Graph and Hypergraph Partitioning

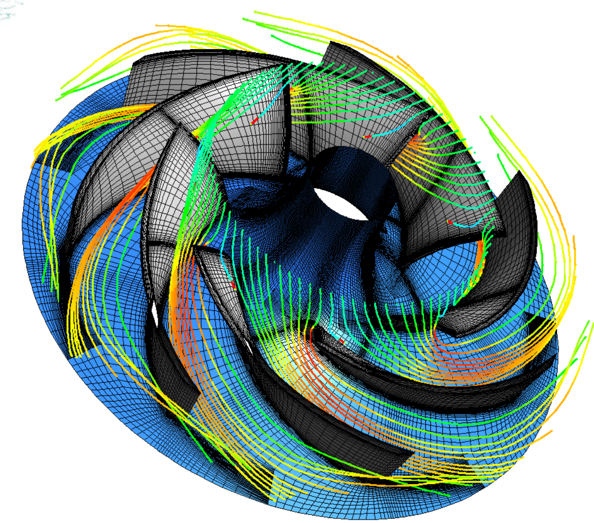
Annual SPP Meeting · October 18, 2017

Yaroslav Akhremtsev, Peter Sanders, Sebastian Schlag, Christian Schulz

INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP



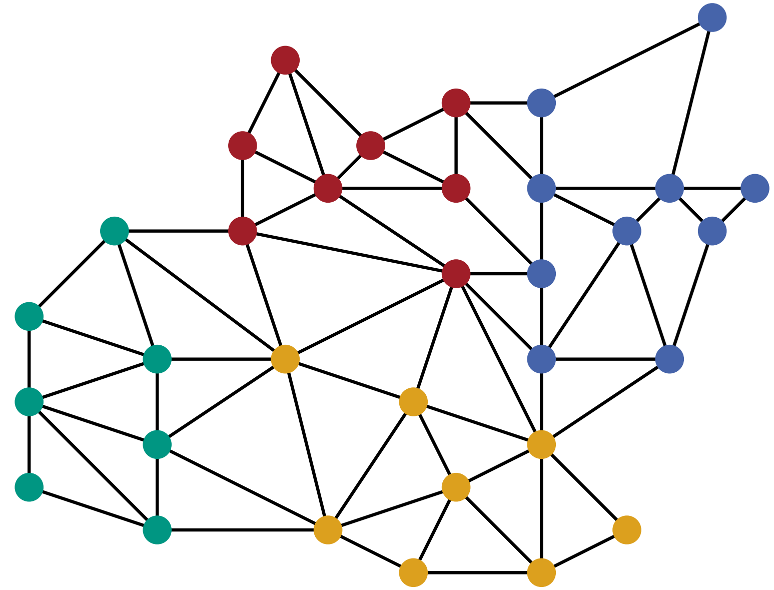




Graphs and Hypergraphs

Graph $G = (V, E)$

vertices → V, edges → E



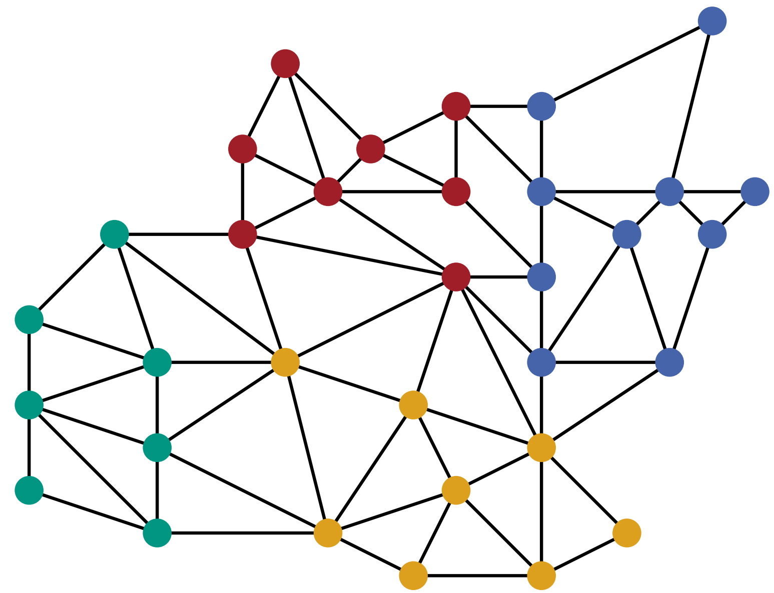
- models **relationships** between **objects**
- dyadic (**2-ary**) relationships

Graphs and Hypergraphs

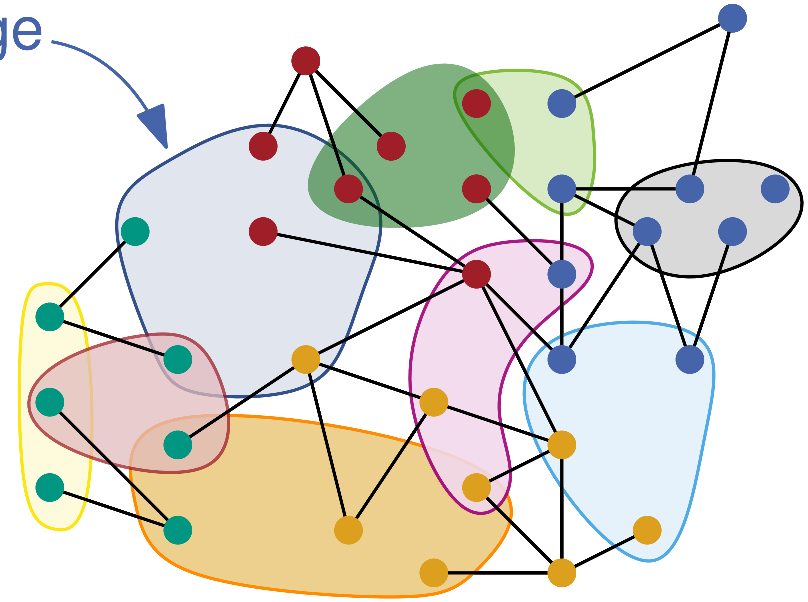
Graph $G = (V, E)$

vertices — edges

- models **relationships** between **objects**
- dyadic (**2-ary**) relationships



hyperedge



Hypergraph $H = (V, E)$

- Generalization of a graph
  $\Rightarrow$ hyperedges connect ≥ 2 nodes
- arbitrary (**d-ary**) relationships
- Edge set $E \subseteq \mathcal{P}(V) \setminus \emptyset$

Institute of Theoretical Informatics
Algorithmics Group

ε-Balanced Graph and Hypergraph Partitioning

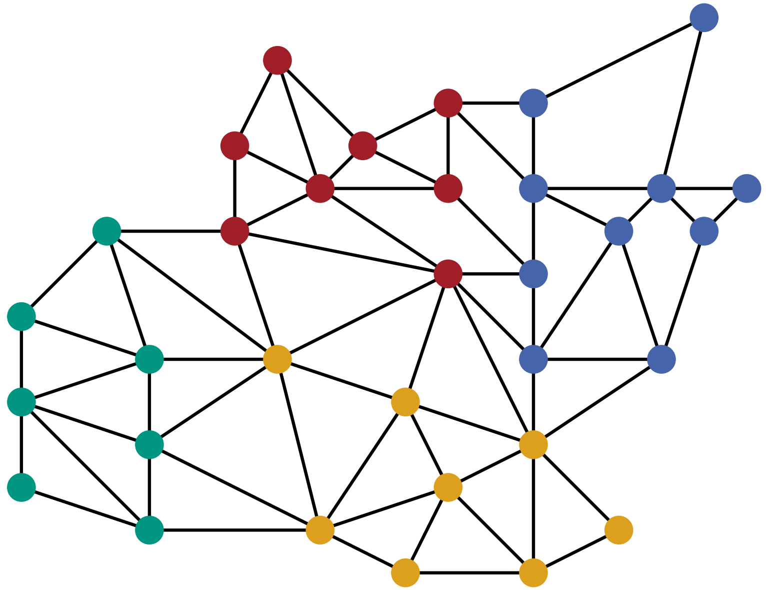
Karlsruhe Institute of Technology

Partition (hyper)graph $G = (V, E, c : V \to \mathbb{R}_{>0}, \omega : E \to \mathbb{R}_{>0})$ into **k** disjoint blocks $V_1, \ldots, V_k$ s.t.

- blocks V_i are **roughly equal-sized**:

$$c(V_i) \leq (1 + \varepsilon) \left\lceil \frac{c(V)}{k} \right\rceil$$

- **objective** function on edges is **minimized**



ε-Balanced Graph and Hypergraph Partitioning

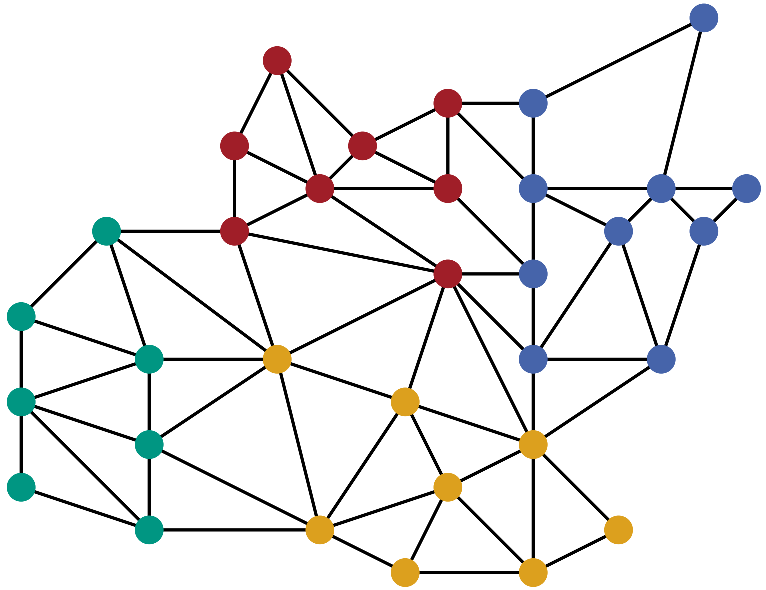
Partition (hyper)graph $G = (V, E, c : V \to \mathbb{R}_{>0}, \omega : E \to \mathbb{R}_{>0})$ into **k** disjoint blocks $V_1, \ldots, V_k$ s.t.

- blocks V_i are **roughly equal-sized**:

imbalance parameter

$$c(V_i) \leq (1 + \varepsilon) \left\lceil \frac{c(V)}{k} \right\rceil$$

- **objective** function on edges is **minimized**



ε-Balanced Graph and Hypergraph Partitioning

Partition (hyper)graph $G = (V, E, c : V \to R_{>0}, \omega : E \to R_{>0})$ into **k** disjoint blocks $V_1, \ldots, V_k$ s.t.

- blocks V_i are **roughly equal-sized**:

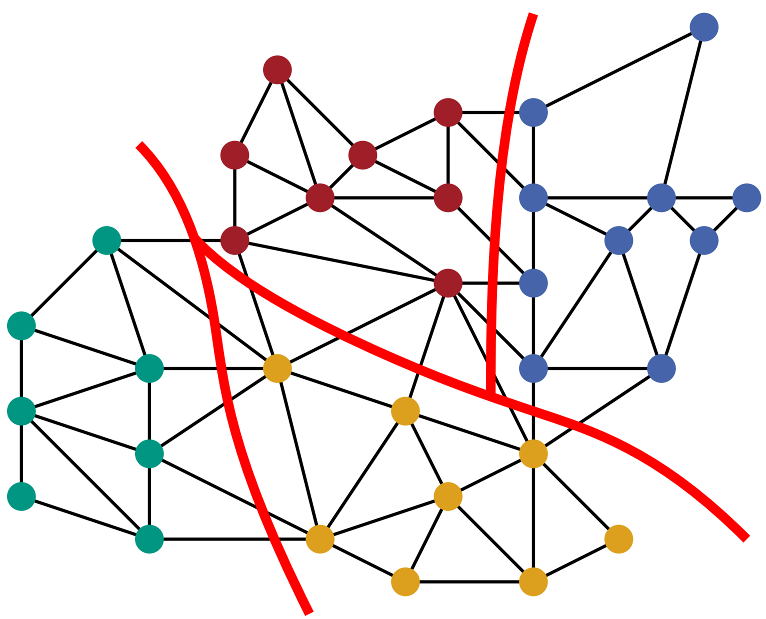
imbalance parameter

$$c(V_i) \leq (1 + \varepsilon) \left\lceil \frac{c(V)}{k} \right\rceil$$

- **objective** function on edges is **minimized**

Common Objectives:

- Graphs:
  - cut: $\sum_{e \in \text{cut}} \omega(e)$



ε-Balanced Graph and Hypergraph Partitioning

Partition (hyper)graph $G = (V, E, c : V \to \mathbb{R}_{>0}, \omega : E \to \mathbb{R}_{>0})$ into **k** disjoint blocks $V_1, \ldots, V_k$ s.t.

- blocks V_i are **roughly equal-sized**:

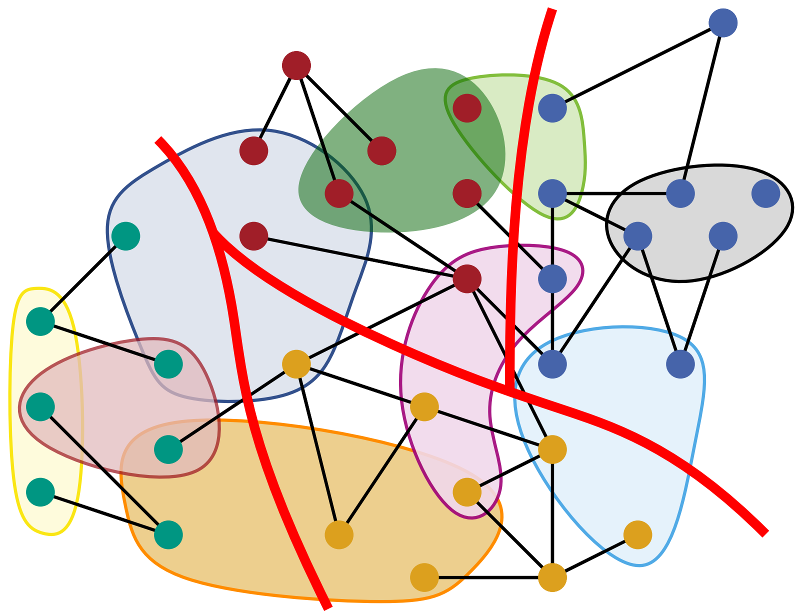
imbalance parameter

$$c(V_i) \leq (1 + \varepsilon) \left\lceil \frac{c(V)}{k} \right\rceil$$

- **objective** function on edges is **minimized**

Common Objectives:

- Graphs:
  - cut: $\sum_{e \in \text{cut}} \omega(e)$



ε-Balanced Graph and Hypergraph Partitioning

Partition (hyper)graph $G = (V, E, c : V \to \mathbb{R}_{>0}, \omega : E \to \mathbb{R}_{>0})$ into **k** disjoint blocks $V_1, \ldots, V_k$ s.t.

- blocks V_i are **roughly equal-sized**:

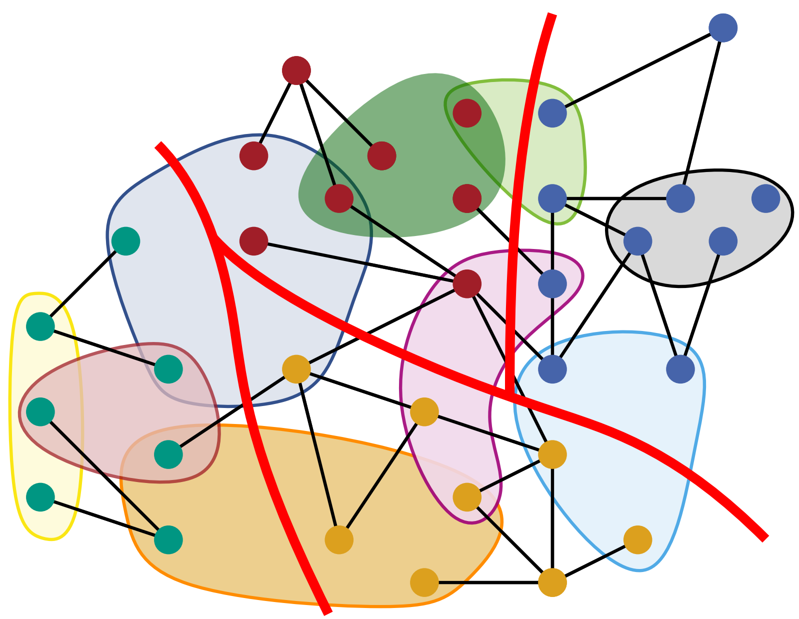
imbalance parameter

$$c(V_i) \leq (1 + \varepsilon) \left\lceil \frac{c(V)}{k} \right\rceil$$

- **objective** function on edges is **minimized**

Common Objectives:

- Graphs:
  - cut: $\sum_{e \in \text{cut}} \omega(e)$
- Hypergraphs:
  - cut: $\sum_{e \in \text{cut}} \omega(e)$



ε-Balanced Graph and Hypergraph Partitioning

Partition (hyper)graph $G = (V, E, c : V \to \mathbb{R}_{>0}, \omega : E \to \mathbb{R}_{>0})$ into **k** disjoint blocks $V_1, \ldots, V_k$ s.t.

- blocks V_i are **roughly equal-sized**:

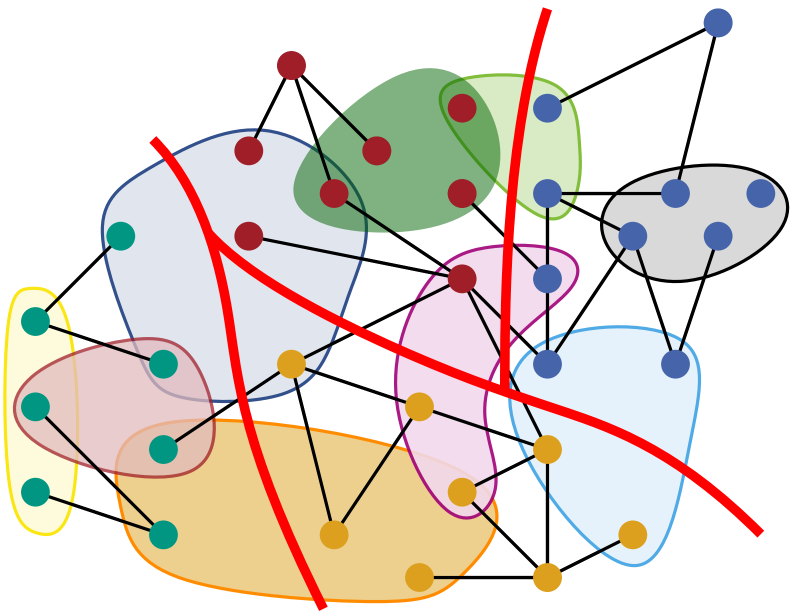
imbalance parameter

$$c(V_i) \leq (1 + \varepsilon) \left\lceil \frac{c(V)}{k} \right\rceil$$

- **objective** function on edges is **minimized**

Common Objectives:

- Graphs:
  - cut: $\sum_{e \in \text{cut}} \omega(e)$
- Hypergraphs:
  - cut: $\sum_{e \in \text{cut}} \omega(e)$
  - connectivity: $\sum_{e \in \text{cut}} (\lambda - 1)\, \omega(e)$



ε-Balanced Graph and Hypergraph Partitioning

Partition (hyper)graph $G = (V, E, c : V \to \mathbb{R}_{>0}, \omega : E \to \mathbb{R}_{>0})$ into **k** disjoint blocks $V_1, \ldots, V_k$ s.t.

- blocks V_i are **roughly equal-sized**:

imbalance parameter

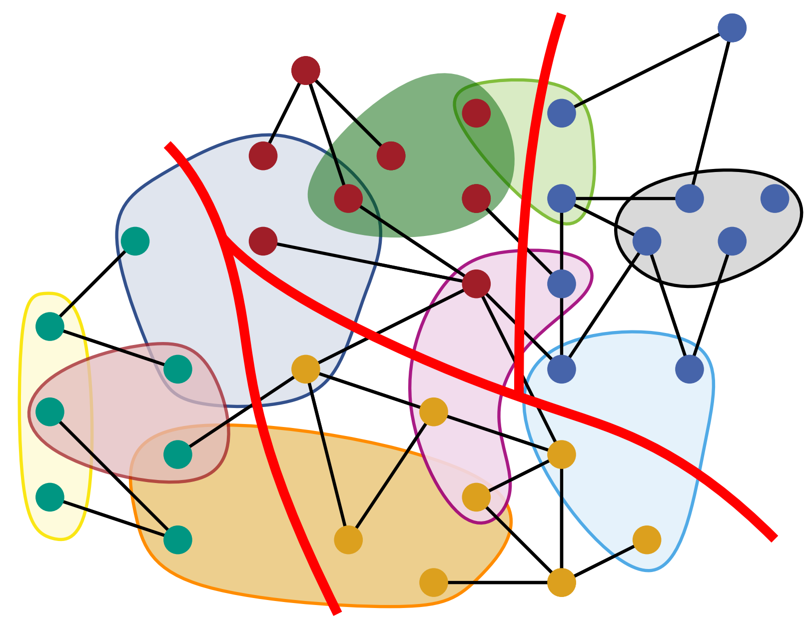
$$c(V_i) \leq (1 + \varepsilon) \left\lceil \frac{c(V)}{k} \right\rceil$$

- **objective** function on edges is **minimized**

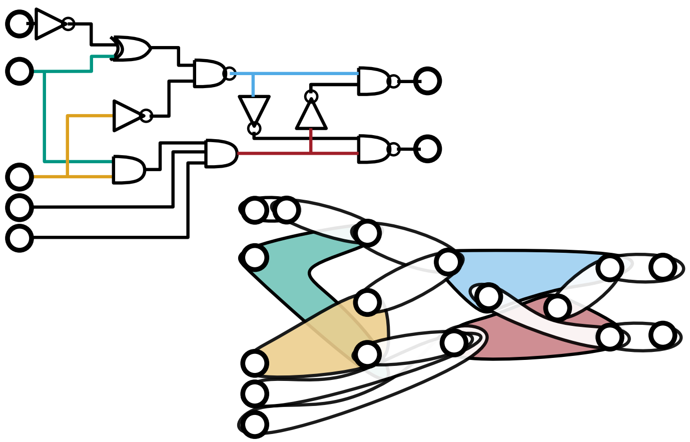
Common Objectives:

- Graphs:
  - cut: $\sum_{e \in \text{cut}} \omega(e)$
- Hypergraphs:
  - cut: $\sum_{e \in \text{cut}} \omega(e)$
  - connectivity: $\sum_{e \in \text{cut}} (\lambda - 1)\, \omega(e)$

 # blocks connected by e



Applications

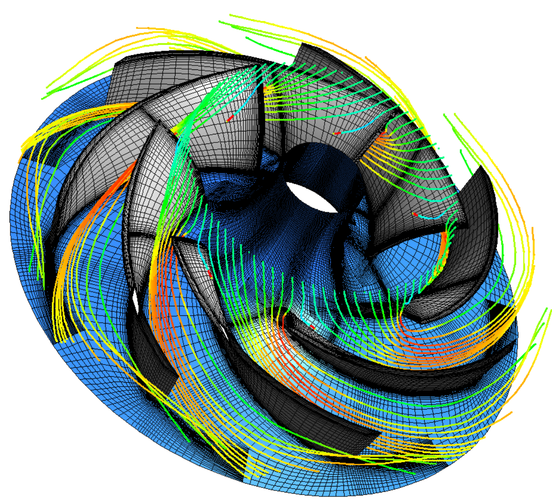


VLSI Design

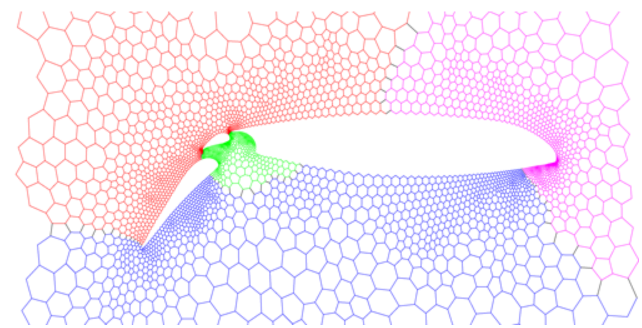
Warehouse Optimization

Complex Networks

Route Planning



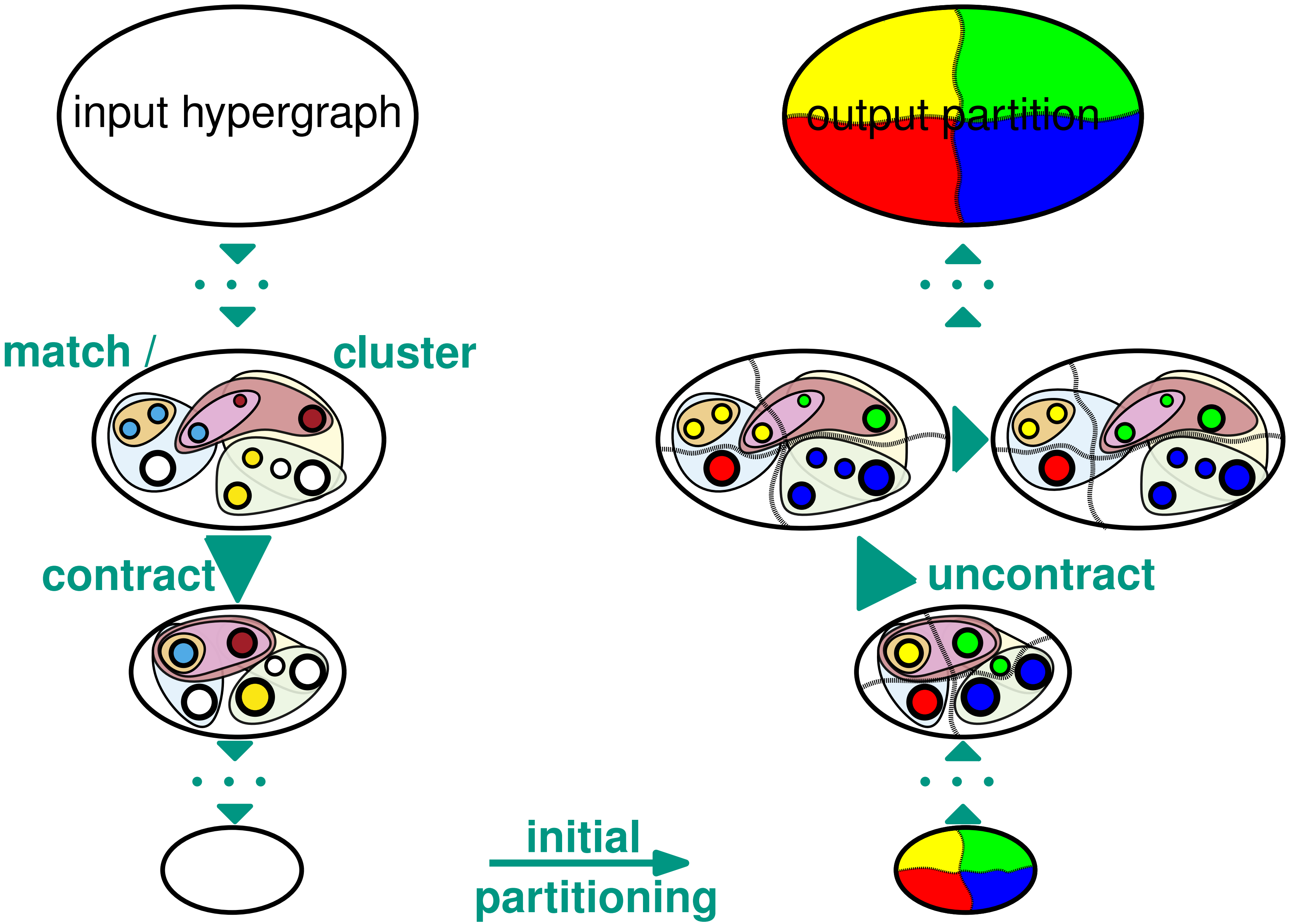
Simulation



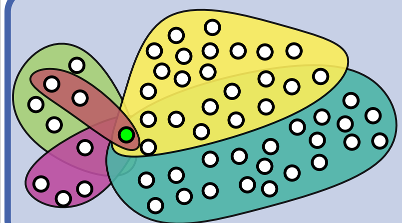
$$\mathbb{R}^{n \times n} \ni Ax = b \in \mathbb{R}^n$$

Scientific Computing

The Multilevel Framework

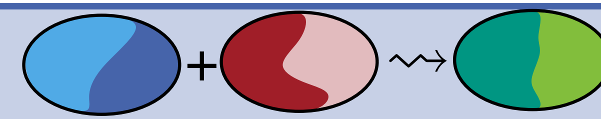


Recent Advances in Hypergraph Partitioning



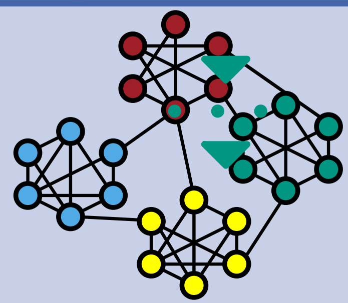
Min-Hash Based Sparsification

Akhremtsev et. al (ALENEX'17)



Memetic Multilevel Algorithm

Andre, Schlag, Schulz (arXiv)



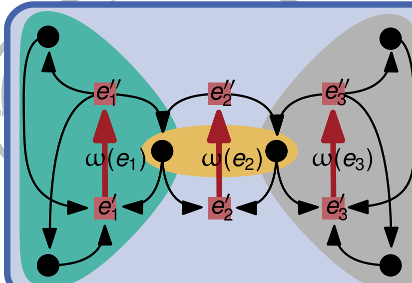
Community-Aware Coarsening

Heuer, Schlag (SEA'17)

Algorithm $A \leftarrow \begin{cases} \text{Config } \mathcal{C}_1 \\ \text{Config } \mathcal{C}_2 \end{cases}$

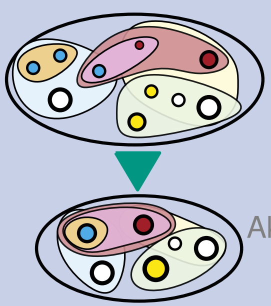
Algorithm Configuration

Öhl, Bachelor's Thesis (upcoming)



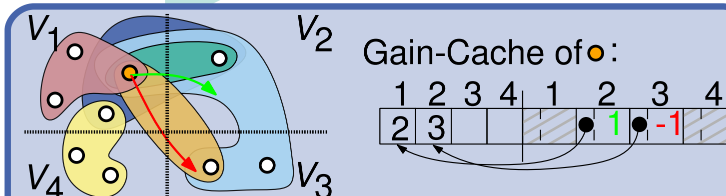
Max-Flow Min-Cut Refinement

Heuer, Master's Thesis (upcoming)



Fast *n*-Level Coarsening

Akhremtsev et. al (ALENEX'17)



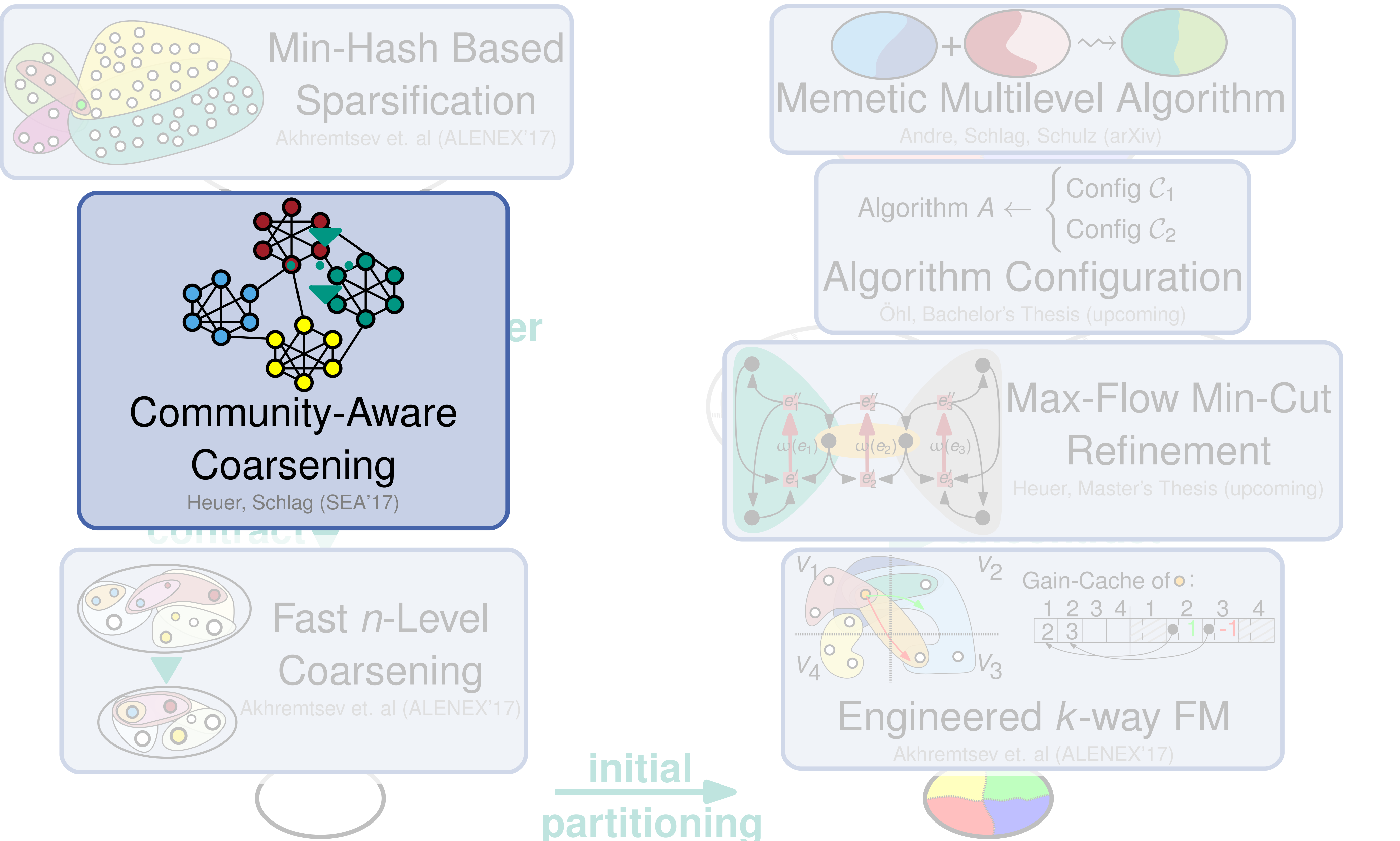
Engineered *k*-way FM

Akhremtsev et. al (ALENEX'17)

initial partitioning

Institute of Theoretical Informatics
Algorithmics Group

Recent Advances in Hypergraph Partitioning



Institute of Theoretical Informatics
Algorithmics Group

Clustering-based Coarsening

Common Strategy: **avoid** global decisions ⇝ **local**, greedy algorithms

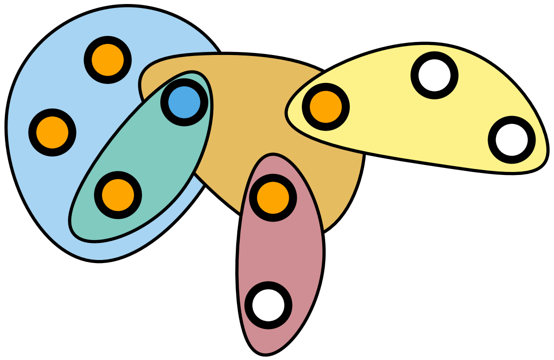
Objective: identify highly connected vertices

using...

foreach *vertex* v **do**

cluster[v] := argmax `rating`(v,u)

neighbor u



Clustering-based Coarsening

Common Strategy: **avoid** global decisions ⇝ **local**, greedy algorithms

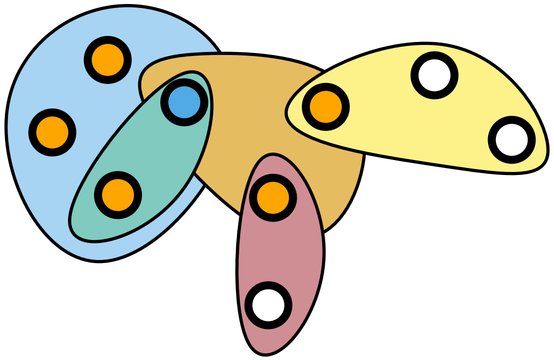
Objective: identify highly connected vertices

using...

foreach *vertex* v **do**

  cluster[v] := argmax `rating`(v,u)

  neighbor u



Main Design Goals: [Karypis,Kumar 99]

1: reduce **size** of nets ⇝ easier local search

2: reduce **number** of nets ⇝ easier initial partitioning

3: maintain **structural similarity** ⇝ good coarse solutions

Clustering-based Coarsening

Common Strategy: **avoid** global decisions ⇝ **local**, greedy algorithms

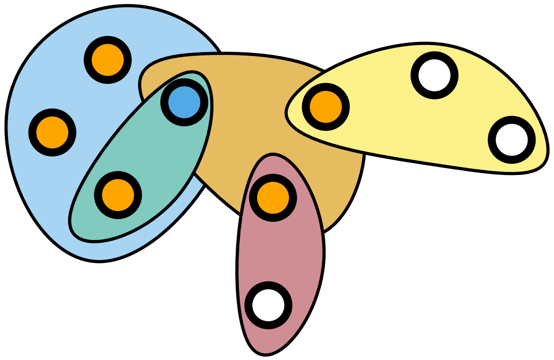
Objective: identify highly connected vertices

using...

foreach *vertex* v **do**

  cluster[v] := argmax `rating`(v,u)

  neighbor u



Main Design Goals: [Karypis,Kumar 99]

1: reduce **size** of nets ⇝ easier local search

2: reduce **number** of nets ⇝ easier initial partitioning

} hypergraph-tailored rating functions ✓

3: maintain **structural similarity** ⇝ good coarse solutions

Clustering-based Coarsening

Common Strategy: **avoid** global decisions ⇝ **local**, greedy algorithms

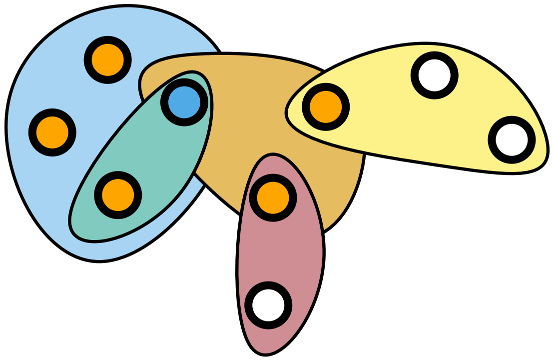
Objective: identify highly connected vertices

using...

foreach *vertex* v **do**

  cluster[v] := argmax `rating`(v,u)

  neighbor u



Main Design Goals: [Karypis,Kumar 99]

1: reduce **size** of nets ⇝ easier local search

2: reduce **number** of nets ⇝ easier initial partitioning

} hypergraph-tailored rating functions ✓

3: maintain **structural similarity** ⇝ good coarse solutions

⟹ prefer clustering over matching

⟹ ensure ∼balanced vertex weights

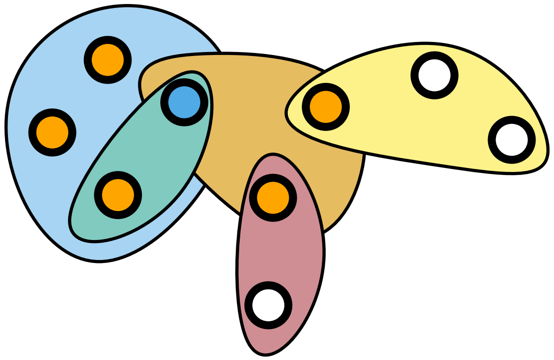
Clustering-based Coarsening

Common Strategy: **avoid** global decisions ⇝ **local**, greedy algorithms

Objective: identify highly connected vertices

using...

foreach *vertex* v **do**
  cluster[v] := argmax `rating`(v,u)
  neighbor u



Main Design Goals: [Karypis,Kumar 99]

1: reduce **size** of nets ⇝ easier local search

2: reduce **number** of nets ⇝ easier initial partitioning

} hypergraph-tailored rating functions ✓

3: maintain **structural similarity** ⇝ good coarse solutions

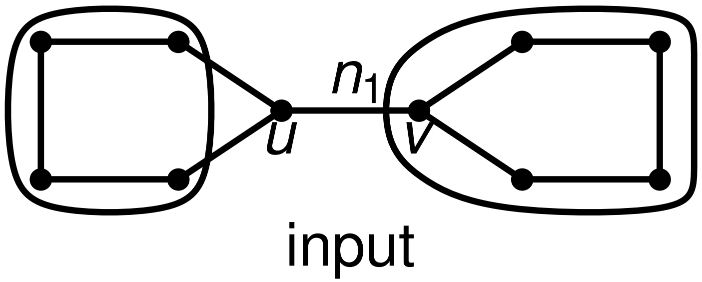
⟹ prefer clustering over matching

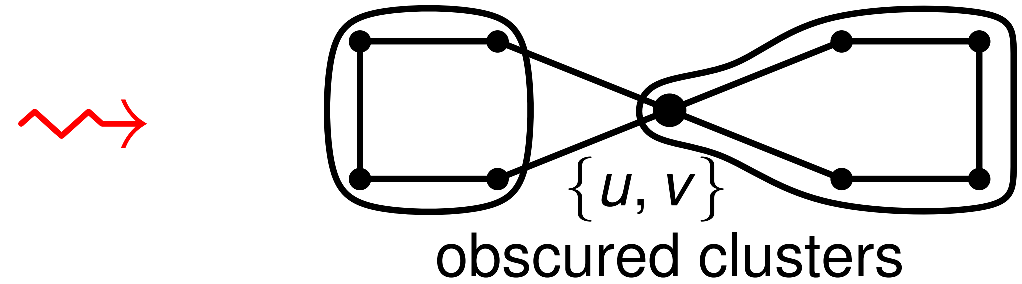
⟹ ensure ∼balanced vertex weights

enough?

What could possibly go wrong?

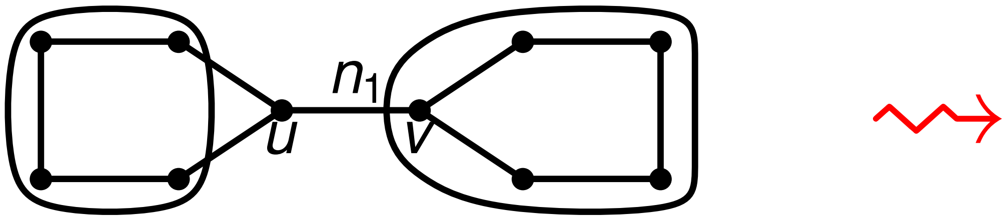
... a lot:



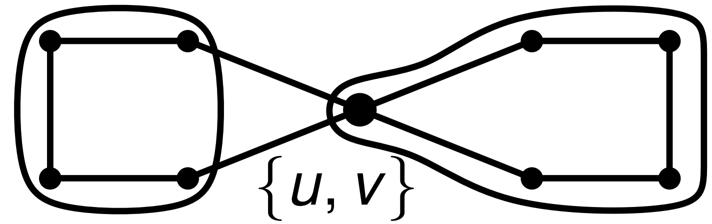


What could possibly go wrong?

... a lot:

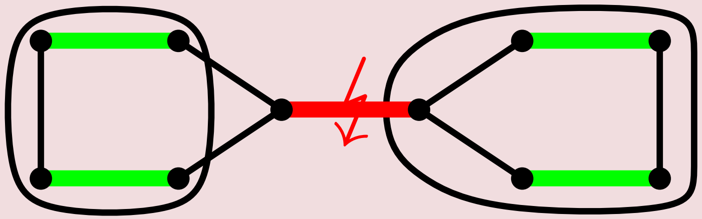


input

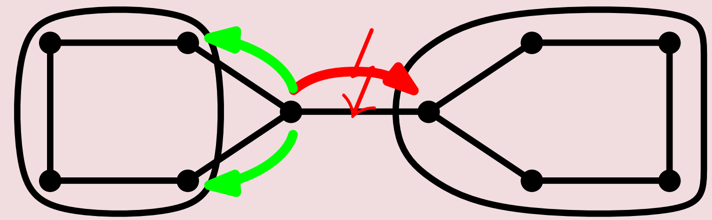


obscured clusters

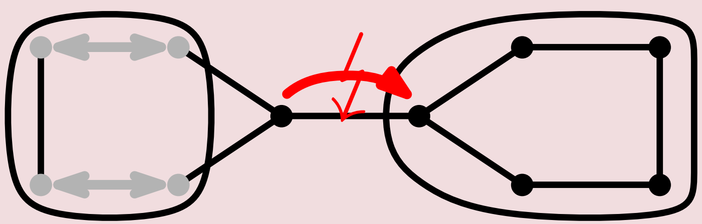
ISSUES



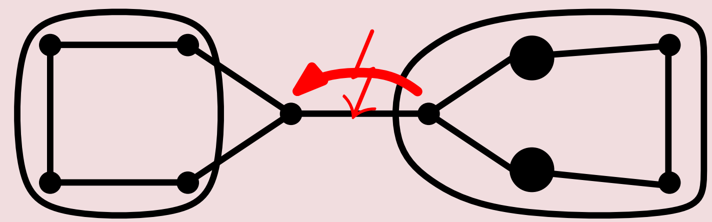
maximal matching



random tie-breaking



prefer unclustered



heavy neighbors

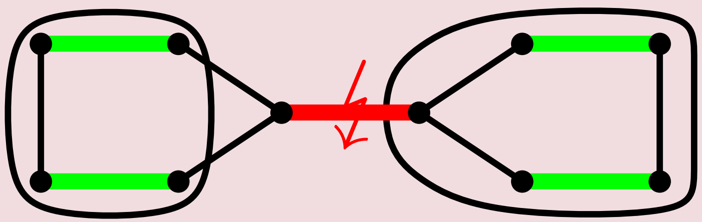
Institute of Theoretical Informatics
Algorithmics Group

What could possibly go wrong?

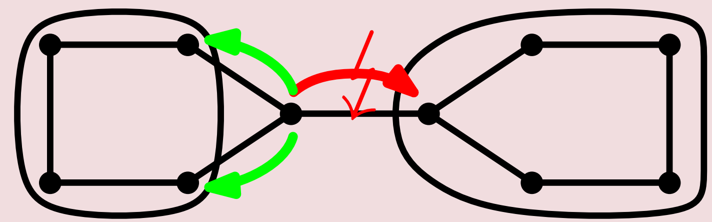
... a lot:

input

obscured clusters

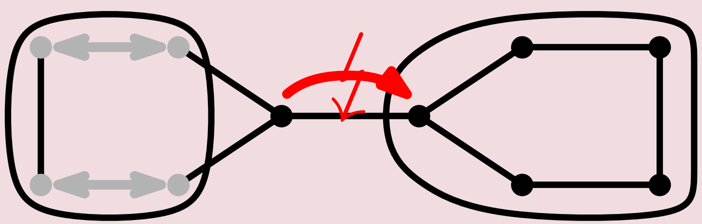


maximal matching

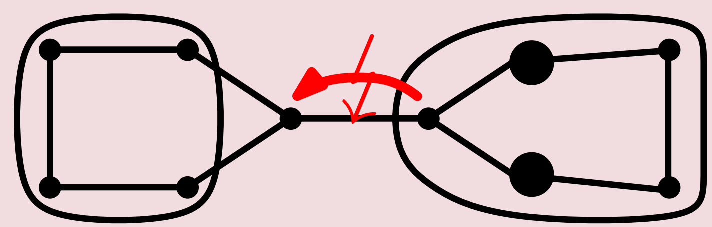


random tie-breaking

ISSUES



prefer unclustered

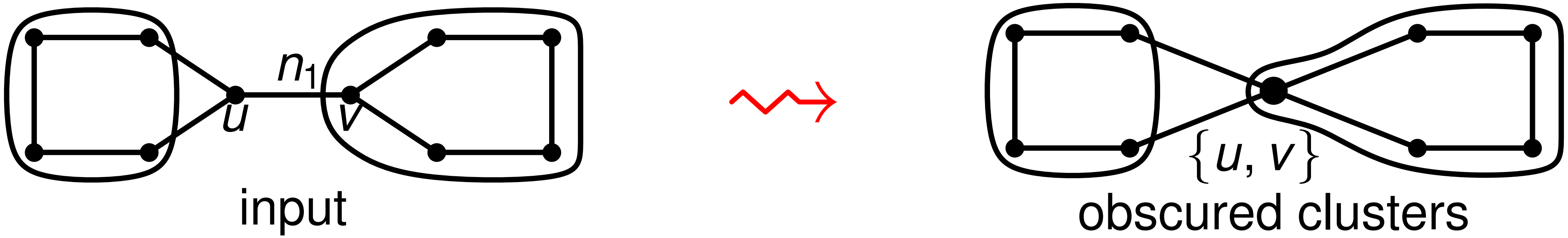


heavy neighbors

⇒ **Problem:** relying **only** on **local** information!

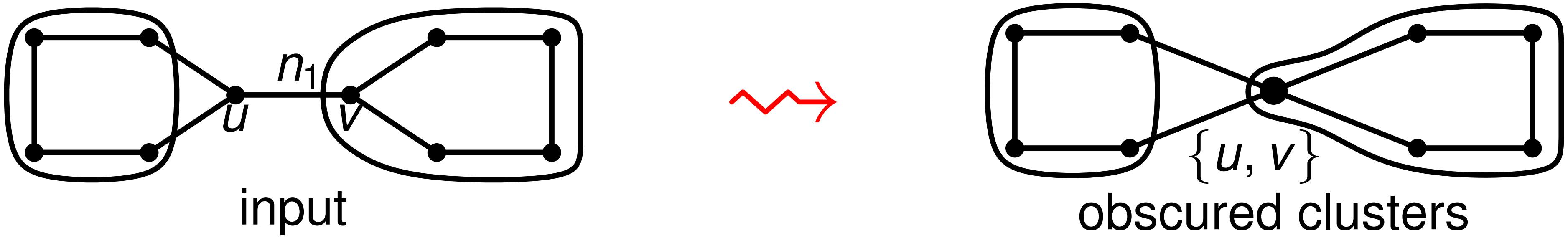
Institute of Theoretical Informatics
Algorithmics Group

Our Approach: Community-aware Coarsening

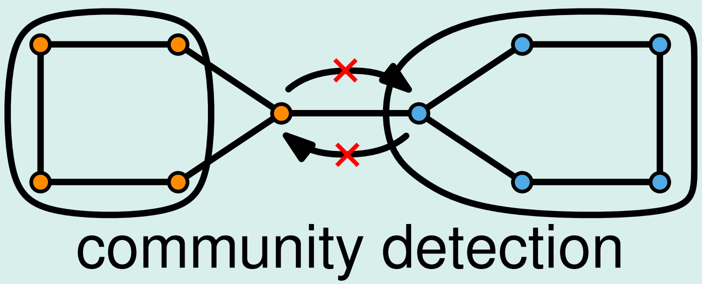


Institute of Theoretical Informatics
Algorithmics Group

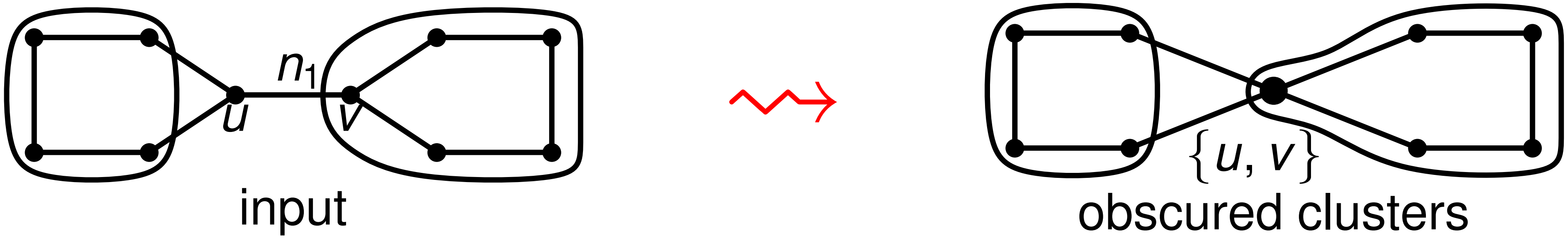
Our Approach: Community-aware Coarsening



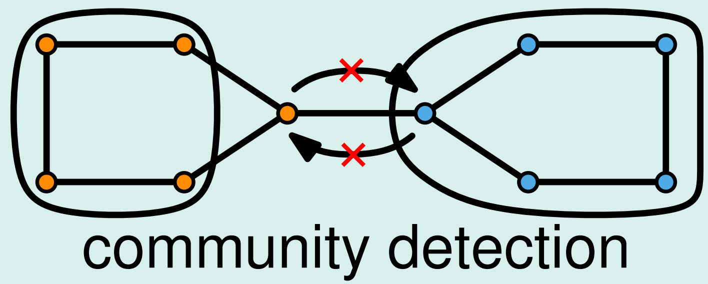
SOLUTION



Our Approach: Community-aware Coarsening

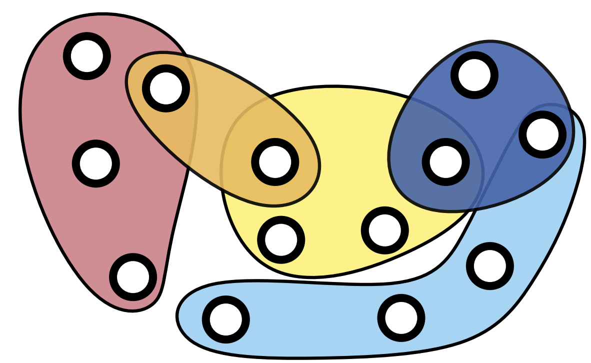


SOLUTION



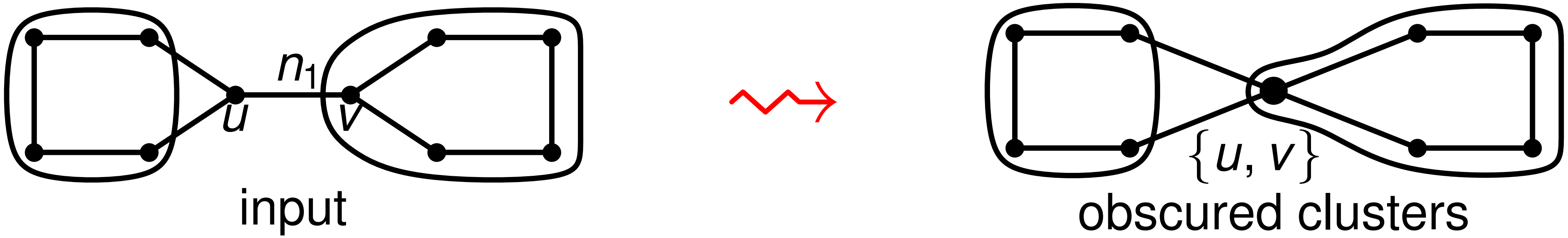
Framework:

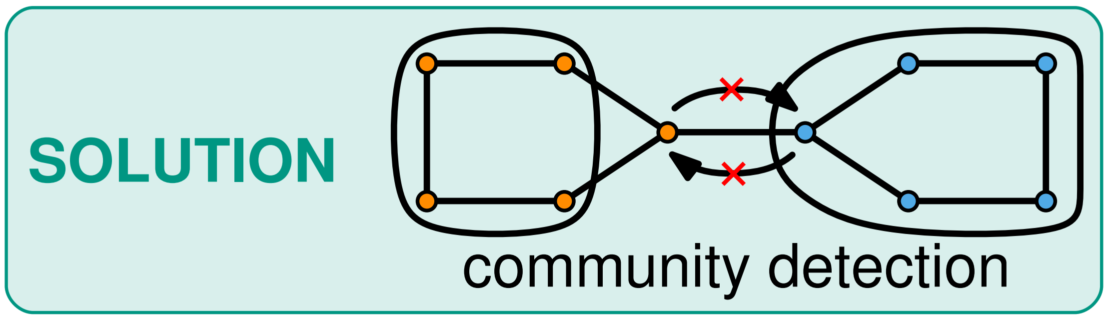
- preprocessing: determine **community structure**
  - using **Louvain** algorithm
  - on **bipartite** graph representation
  - structural properties via **edge weights**
- only allow **intra-community** contractions



Institute of Theoretical Informatics
Algorithmics Group

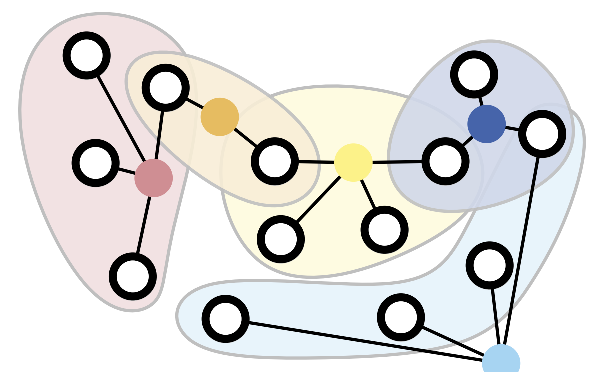
Our Approach: Community-aware Coarsening





Framework:

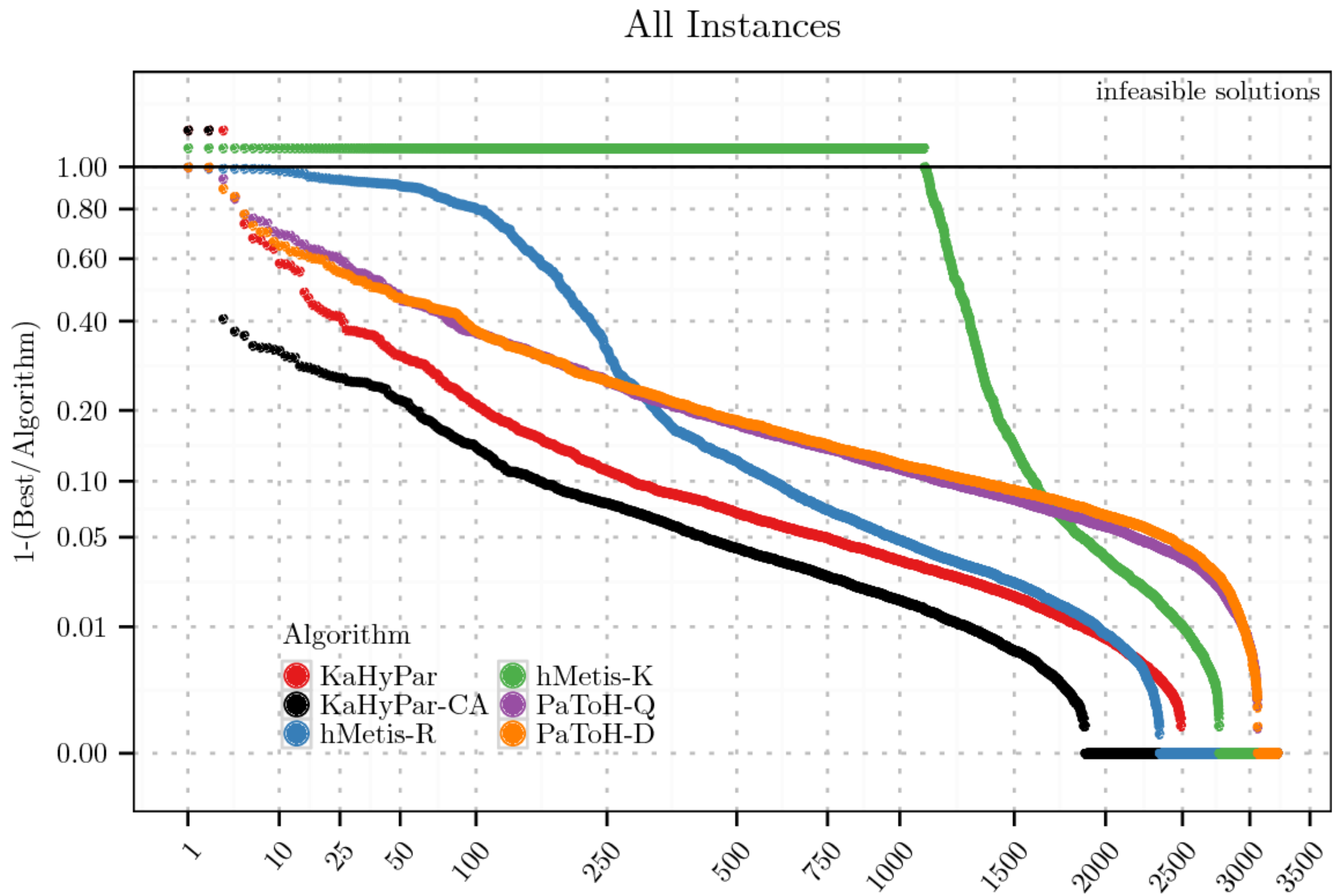
- preprocessing: determine **community structure**
  - using **Louvain** algorithm
  - on **bipartite** graph representation
  - structural properties via **edge weights**
- only allow **intra-community** contractions



Institute of Theoretical Informatics
Algorithmics Group

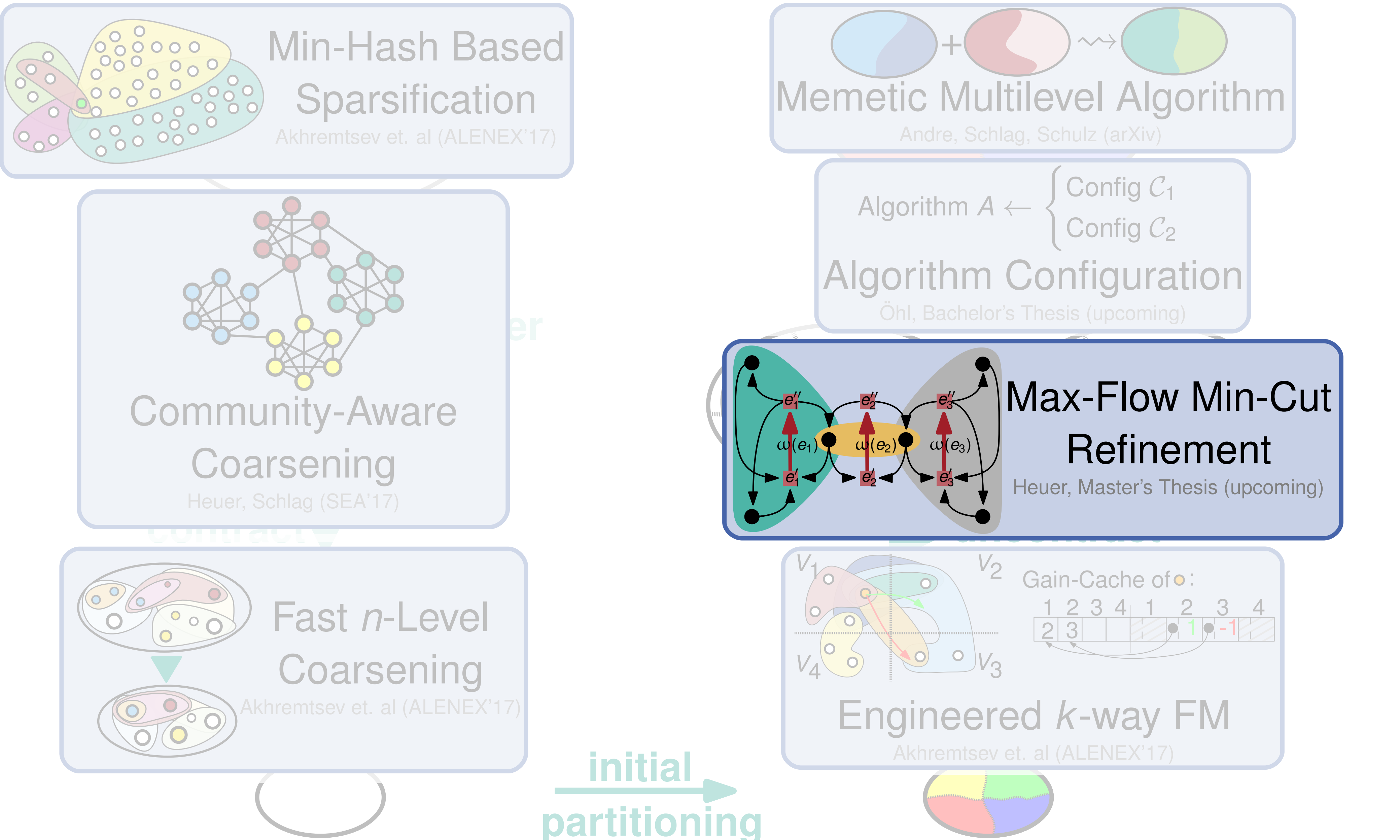
Experimental Results

- 488 hypergraphs (VLSI, UF Sparse Matrix Collection, SAT Competition)
- $k \in \{2, 4, 8, 16, 32, 64, 128\}$ with imbalance: $\varepsilon = 3\%$

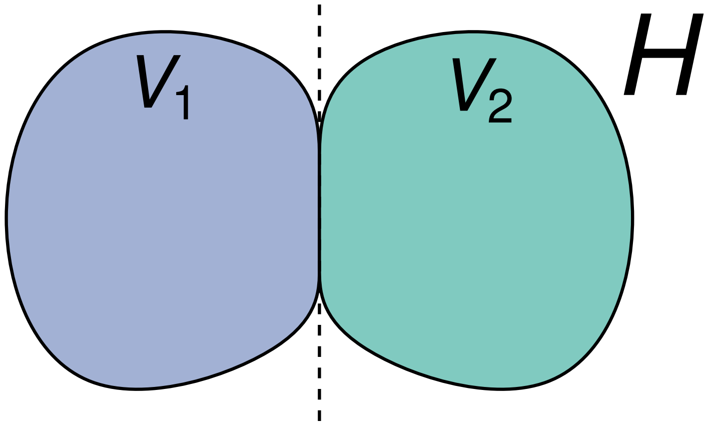


Institute of Theoretical Informatics
Algorithmics Group

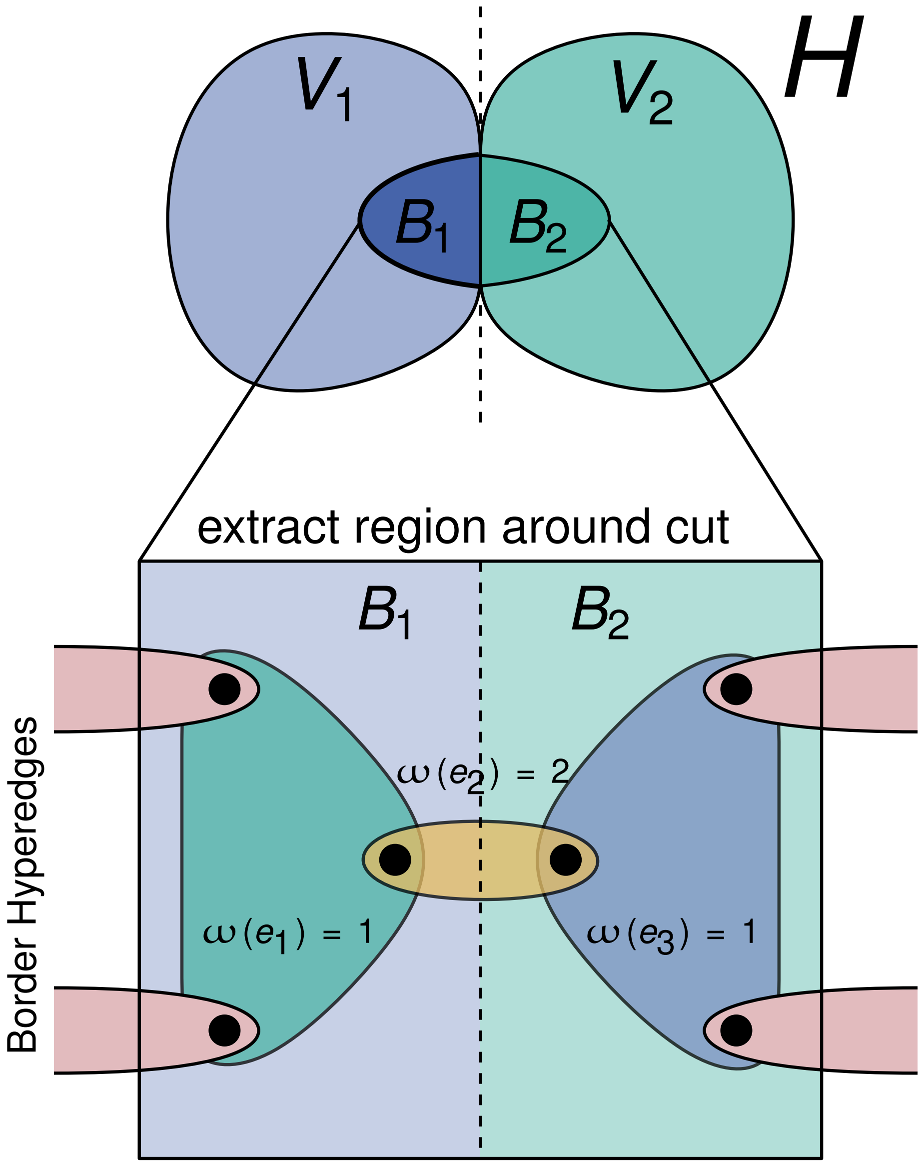
Recent Advances in Hypergraph Partitioning



Max-Flow Min-Cut Refinement

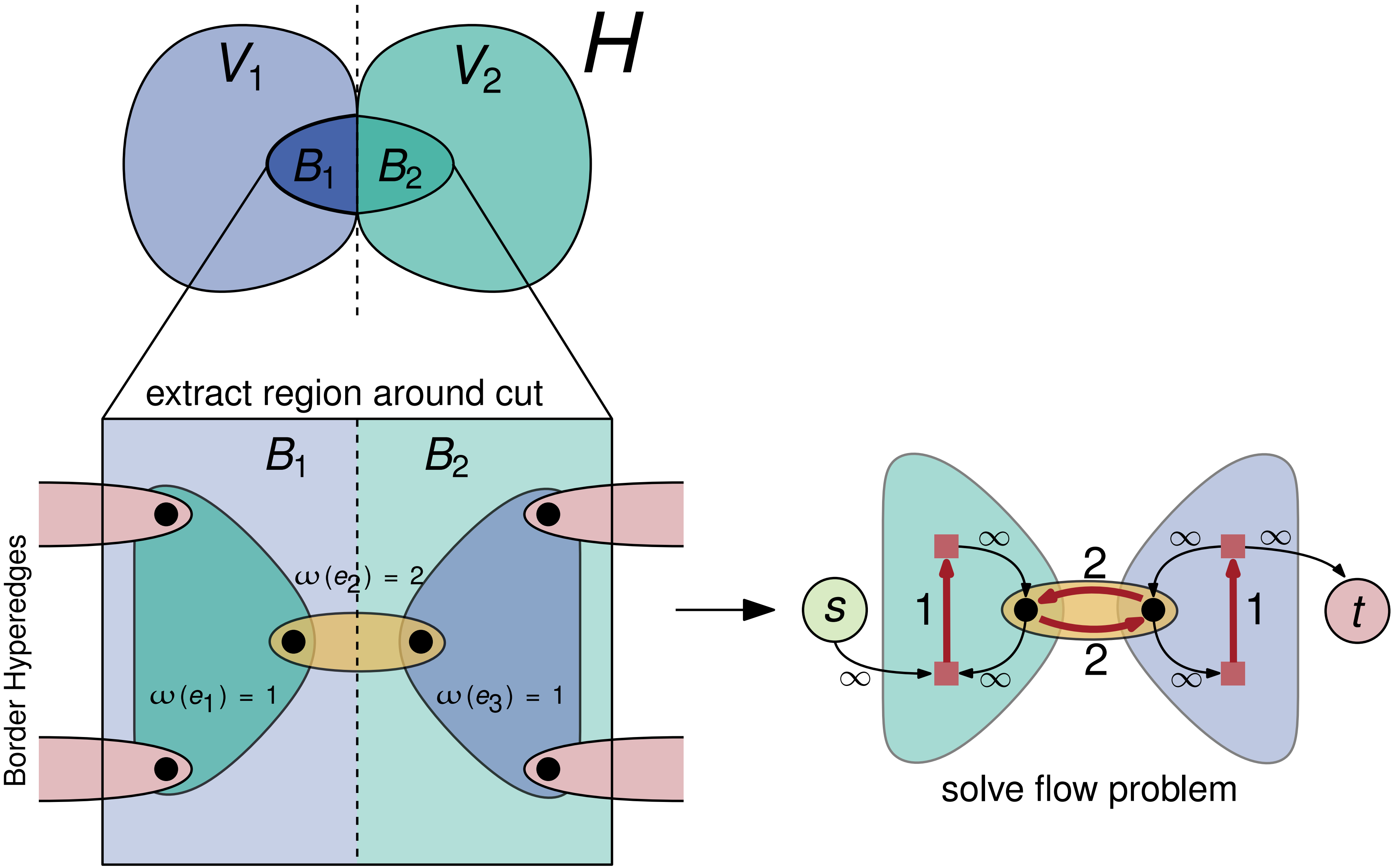


Max-Flow Min-Cut Refinement



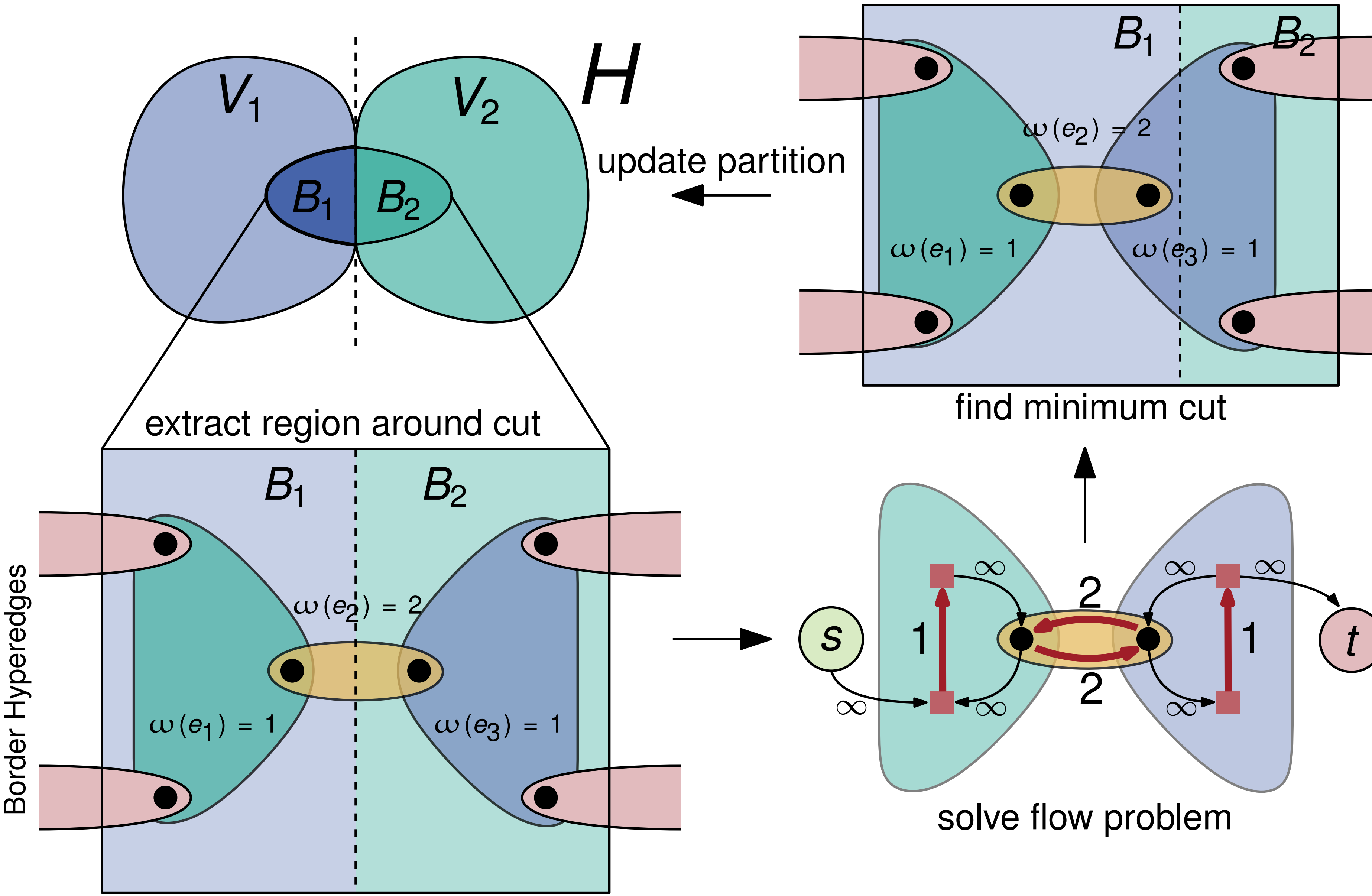
Institute of Theoretical Informatics
Algorithmics Group

Max-Flow Min-Cut Refinement



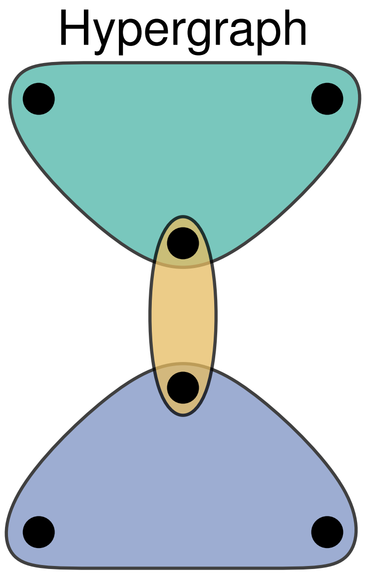
Institute of Theoretical Informatics
Algorithmics Group

Max-Flow Min-Cut Refinement



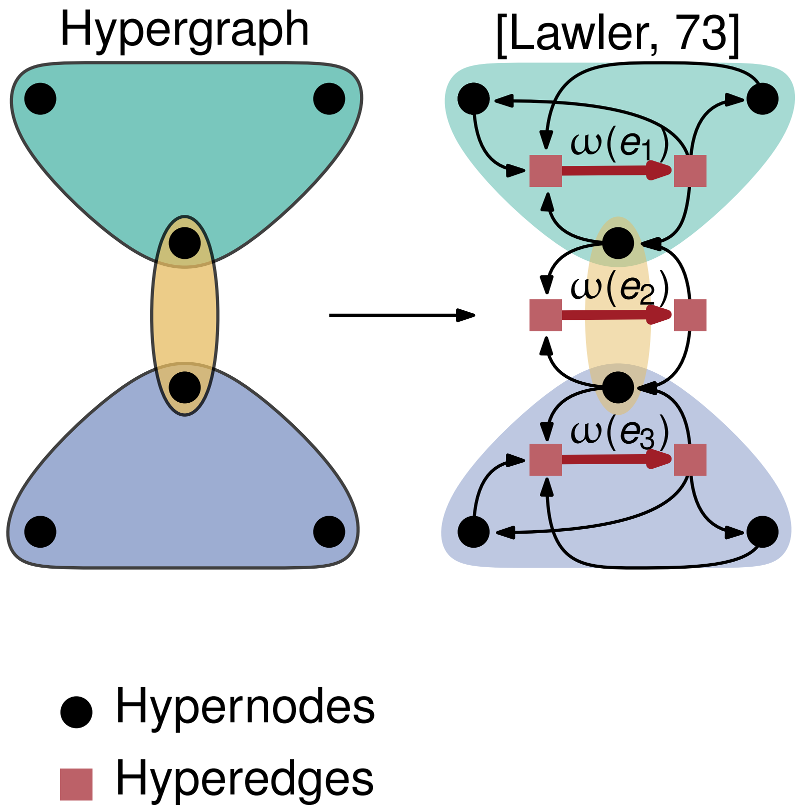
Institute of Theoretical Informatics
Algorithmics Group

From Hypergraphs to Flow-Networks

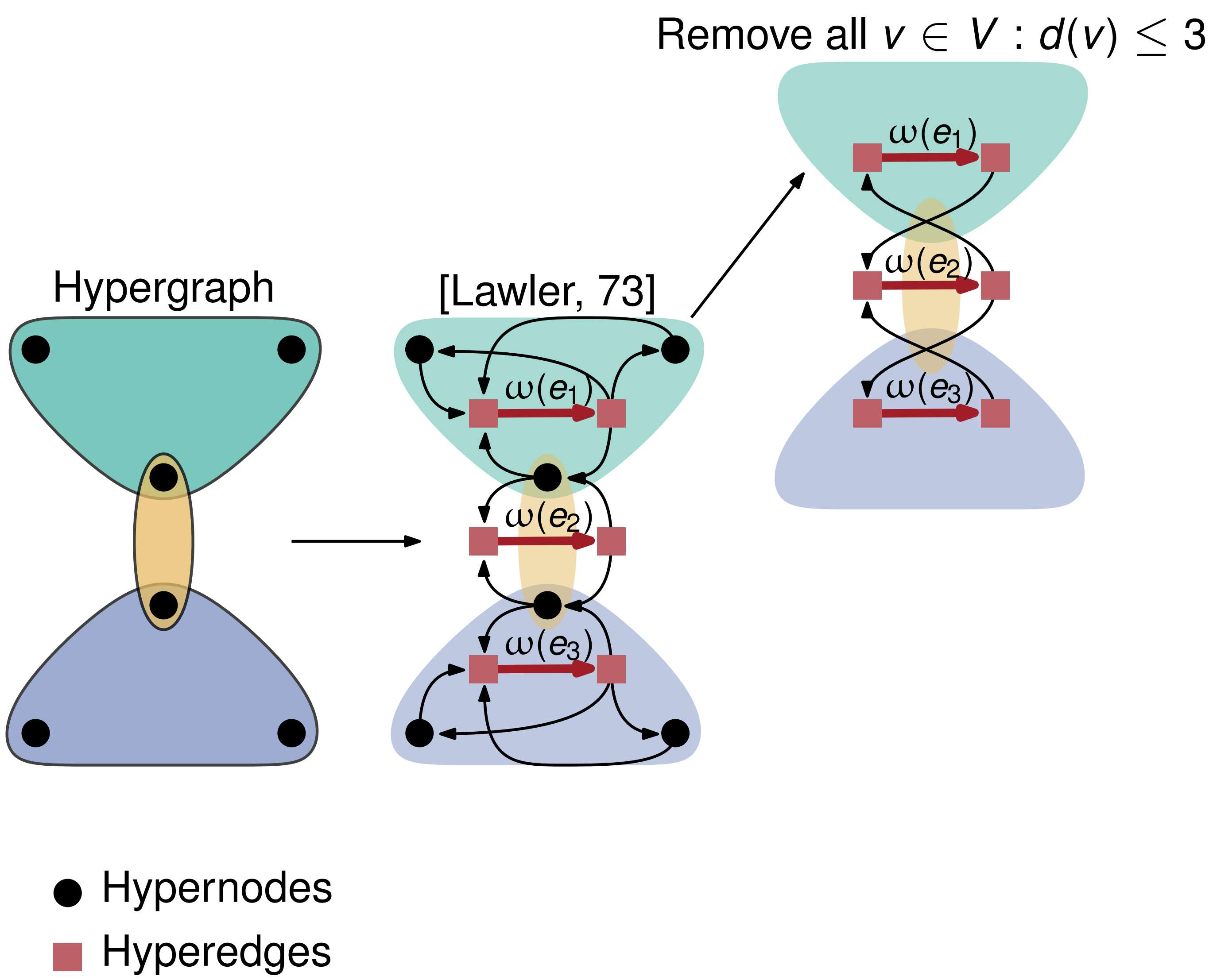


Institute of Theoretical Informatics
Algorithmics Group

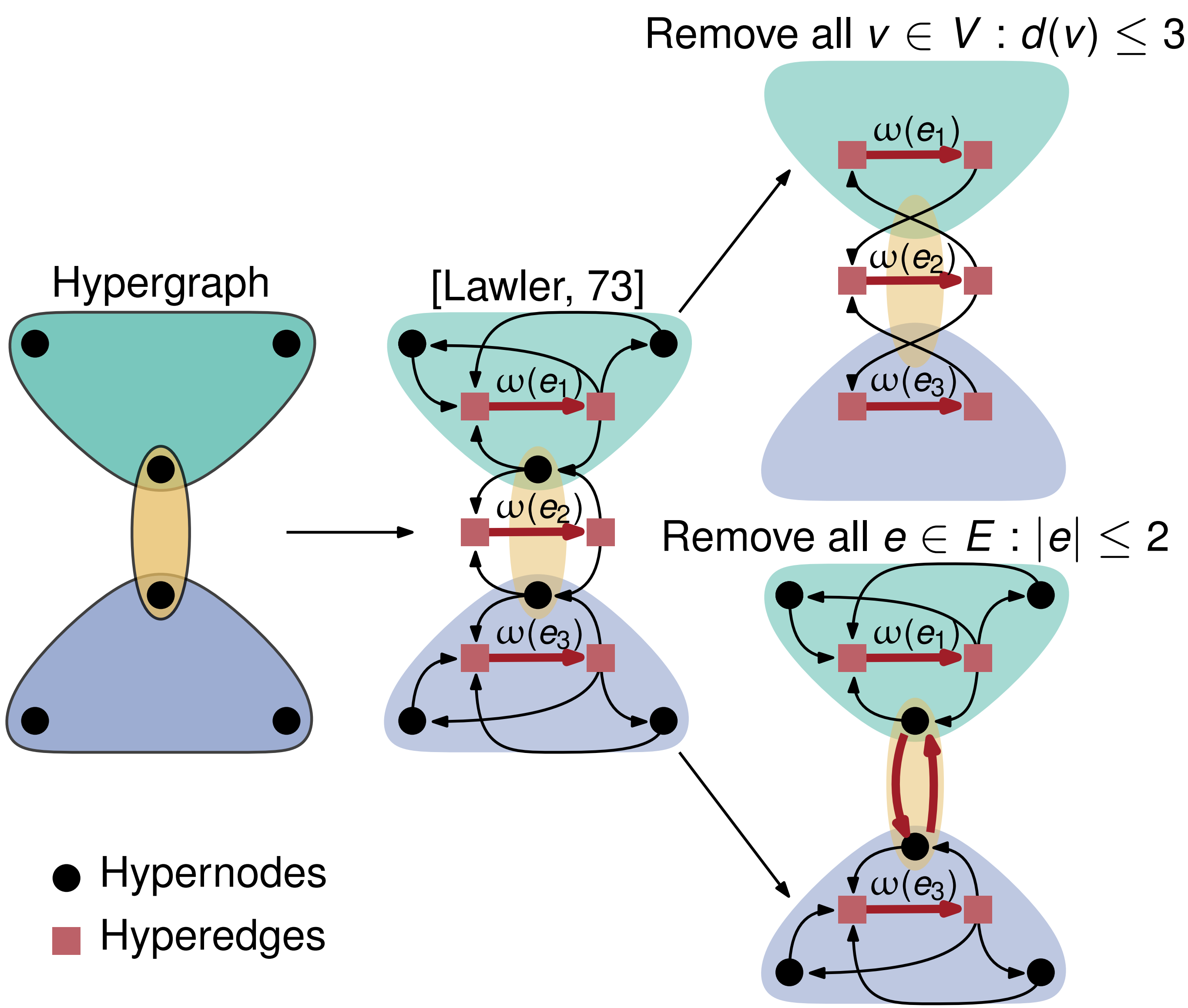
From Hypergraphs to Flow-Networks



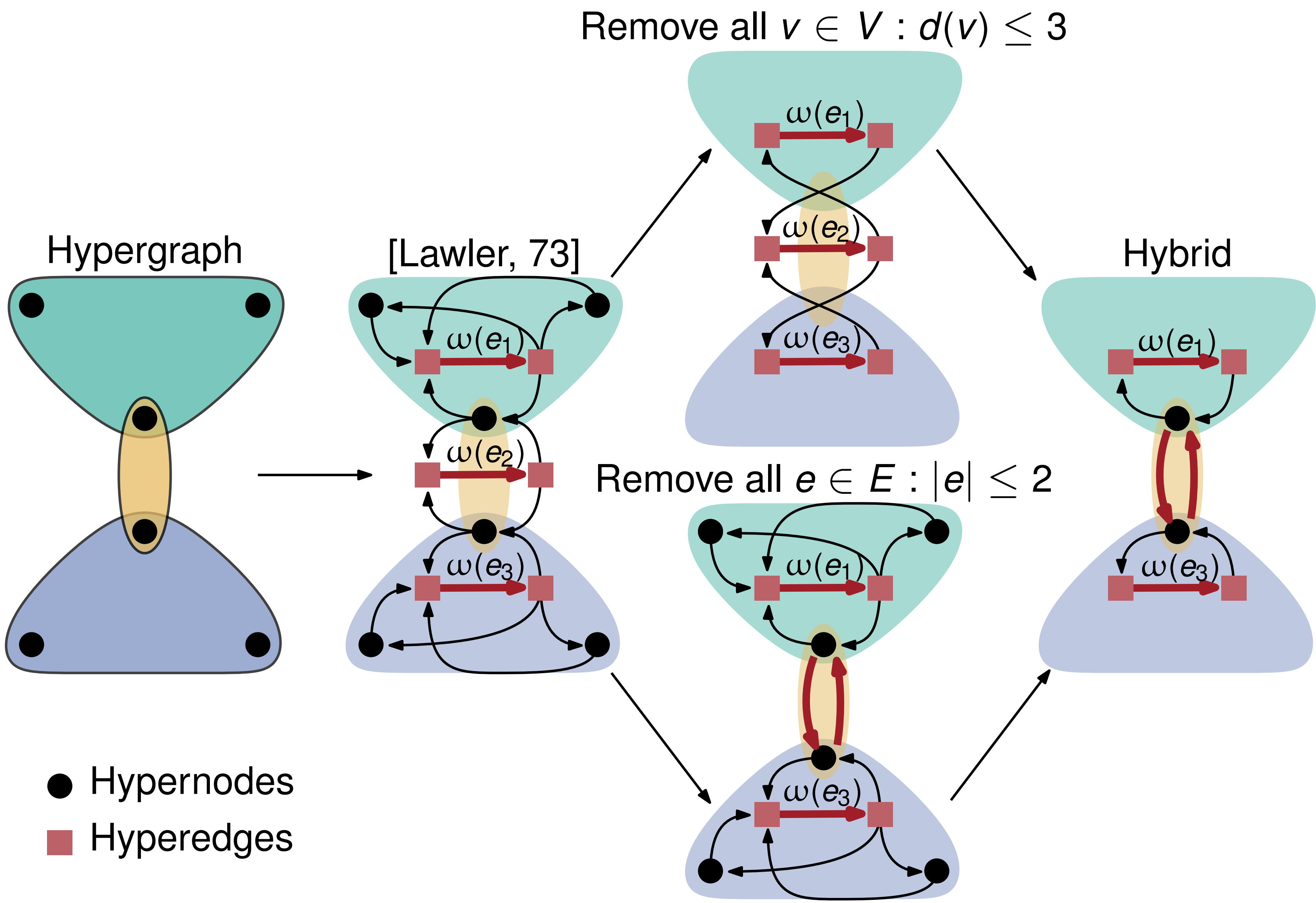
From Hypergraphs to Flow-Networks



From Hypergraphs to Flow-Networks

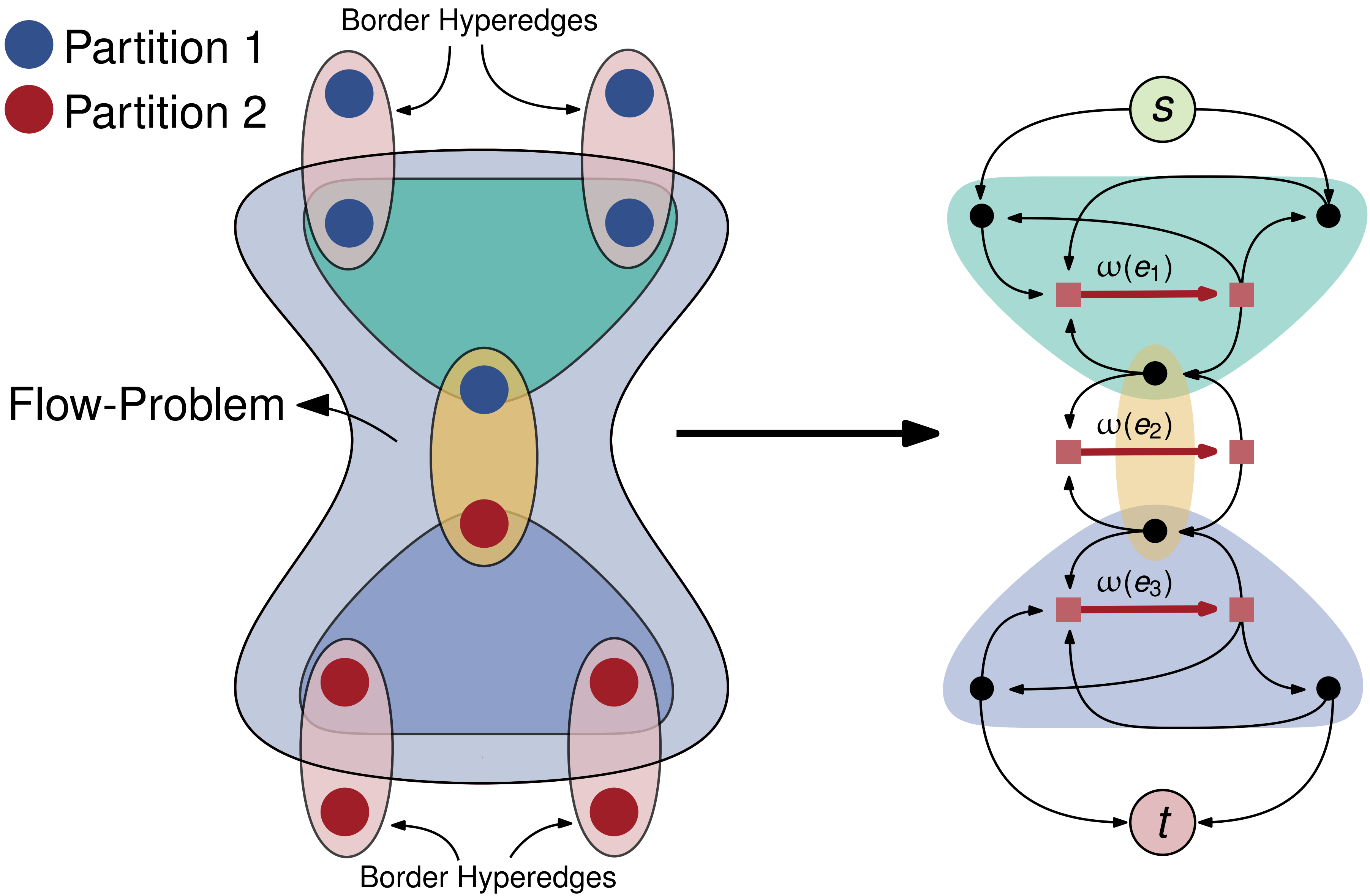


From Hypergraphs to Flow-Networks



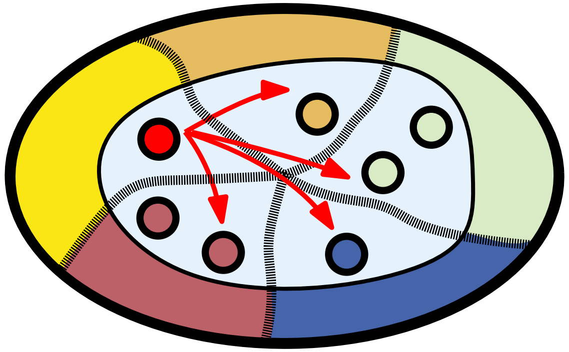
Institute of Theoretical Informatics
Algorithmics Group

Flow Refinement - Modeling Details



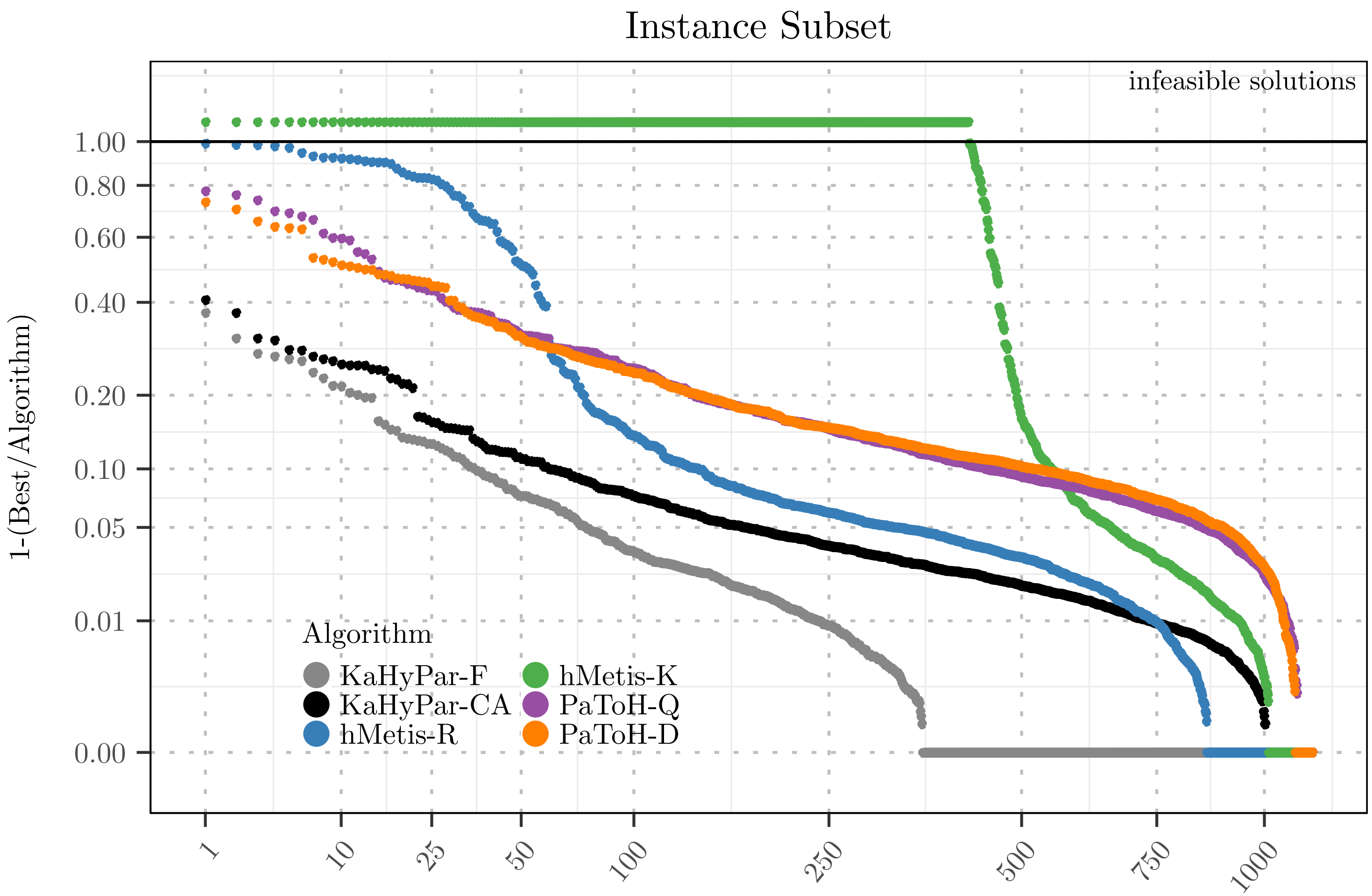
Implementation Details

- *n*-level flow refinement **expensive** $\rightsquigarrow$ emulate multilevel-approach
  $\Rightarrow$ employ every $\frac{n}{\log n}$ uncontractions
- direct *k*-way optimization via **pairwise** flow refinement
- active block scheduling
- most-balanced minimum cuts
- combined with direct *k*-way localized FM local search (every level)



direct *k*-way FM

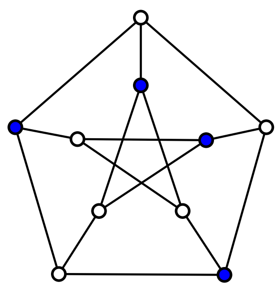
Flow Refinement - Preliminary Results



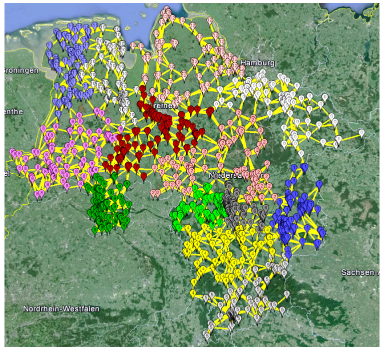
Algorithm	$t[s]$
KaHyPar-CA	21.0
KaHyPar-F	46.8
hMetis-R	63.1

Recent Advances in Graph Partitioning

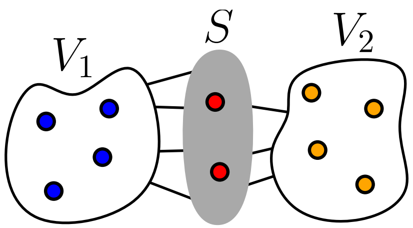
Applications:



Indepentent Sets
Lamm, Sanders, Schulz (SEA'15)
Lamm et. al (ALENEX'16)



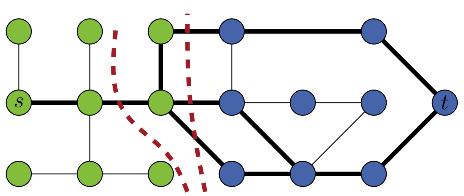
Territory Design
Ahuja et. al (GIS'15)



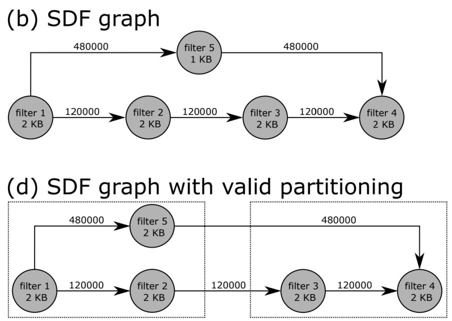
Node Separators
Sanders, Schulz (SEA'16)
Schulz et. al (GECCO'17)

Process Mapping
Schulz, Träff (SEA'17)

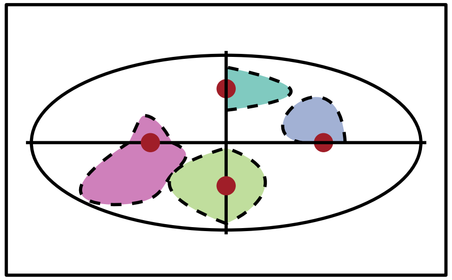
Algorithms:



Flow-Cutter
Hamann, Strasser (ALENEX'16)
Strasser (PACE'16, PACE'17)



DAG Partitioning
Moreia, Popp, Schulz (SEA'17)

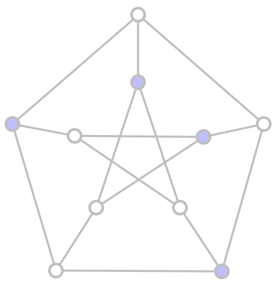


Shared Memory Parallel
Akhremtsev, Sanders, Schulz

Recent Advances in Graph Partitioning

Karlsruhe Institute of Technology

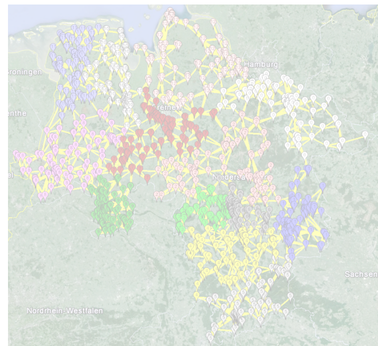
Applications:



Indepentent Sets

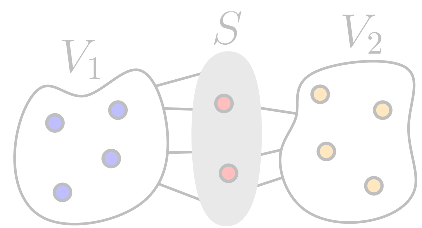
Lamm, Sanders, Schulz (SEA'15)

Lamm et. al (ALENEX'16)



Territory Design

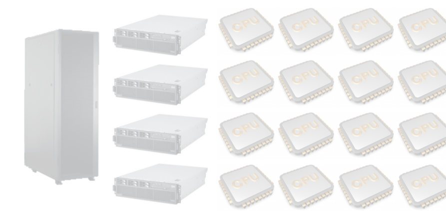
Ahuja et. al (GIS'15)



Node Separators

Sanders, Schulz (SEA'16)

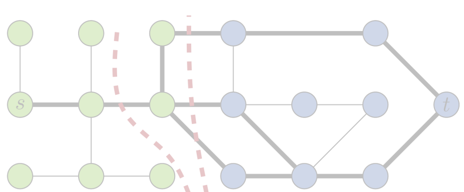
Schulz et. al (GECCO'17)



Process Mapping

Schulz, Träff (SEA'17)

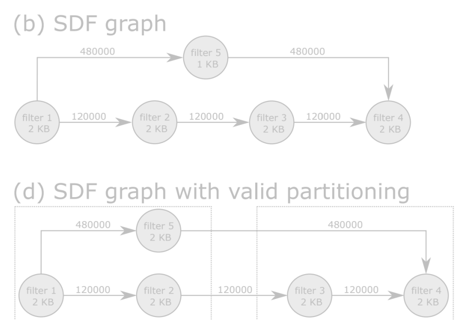
Algorithms:



Flow-Cutter

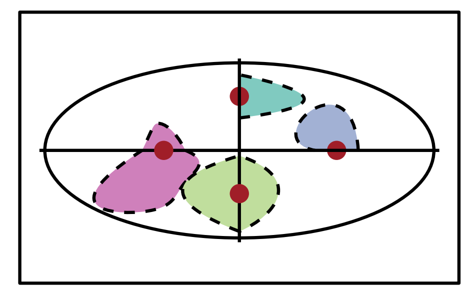
Hamann, Strasser (ALENEX'16)

Strasser (PACE'16, PACE'17)



DAG Partitioning

Moreia, Popp, Schulz (SEA'17)



Shared Memory Parallel

Akhremtsev, Sanders, Schulz

Institute of Theoretical Informatics
Algorithmics Group

Parallelizing KaHIP

Goal: High-quality Shared-Memory Graph Partitioner

- **parallel** coarsening, initial partitioning and refinement phases
- constructs balanced solutions of high quality
- good speed-ups, especially for local search algorithms

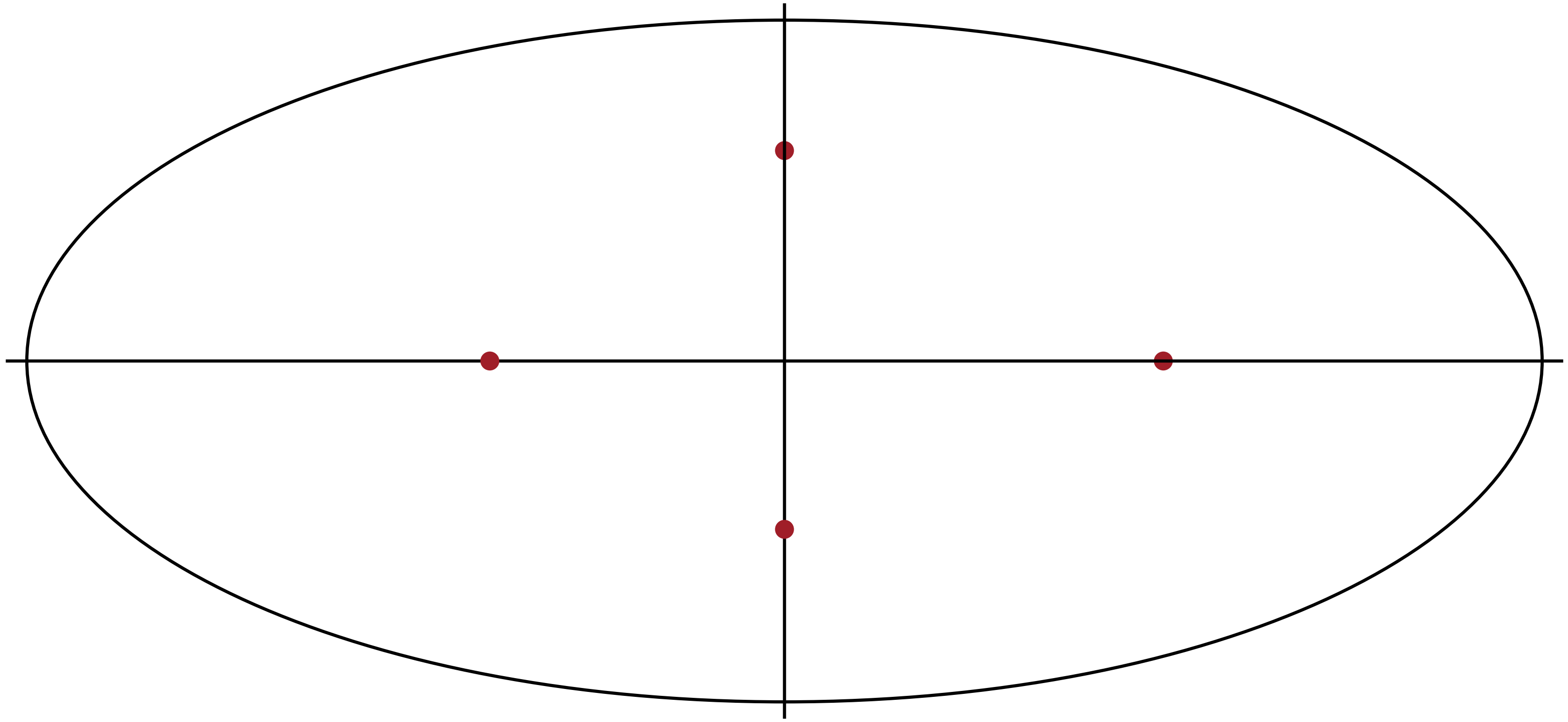
Implementation Details:

- cache-aligned arrays
- TBB scalable allocator
- thread-pinning
- cache-aware hash tables

Slides by Y. Akhremtsev

Parallelizing KaHIP - Parallel Local Search

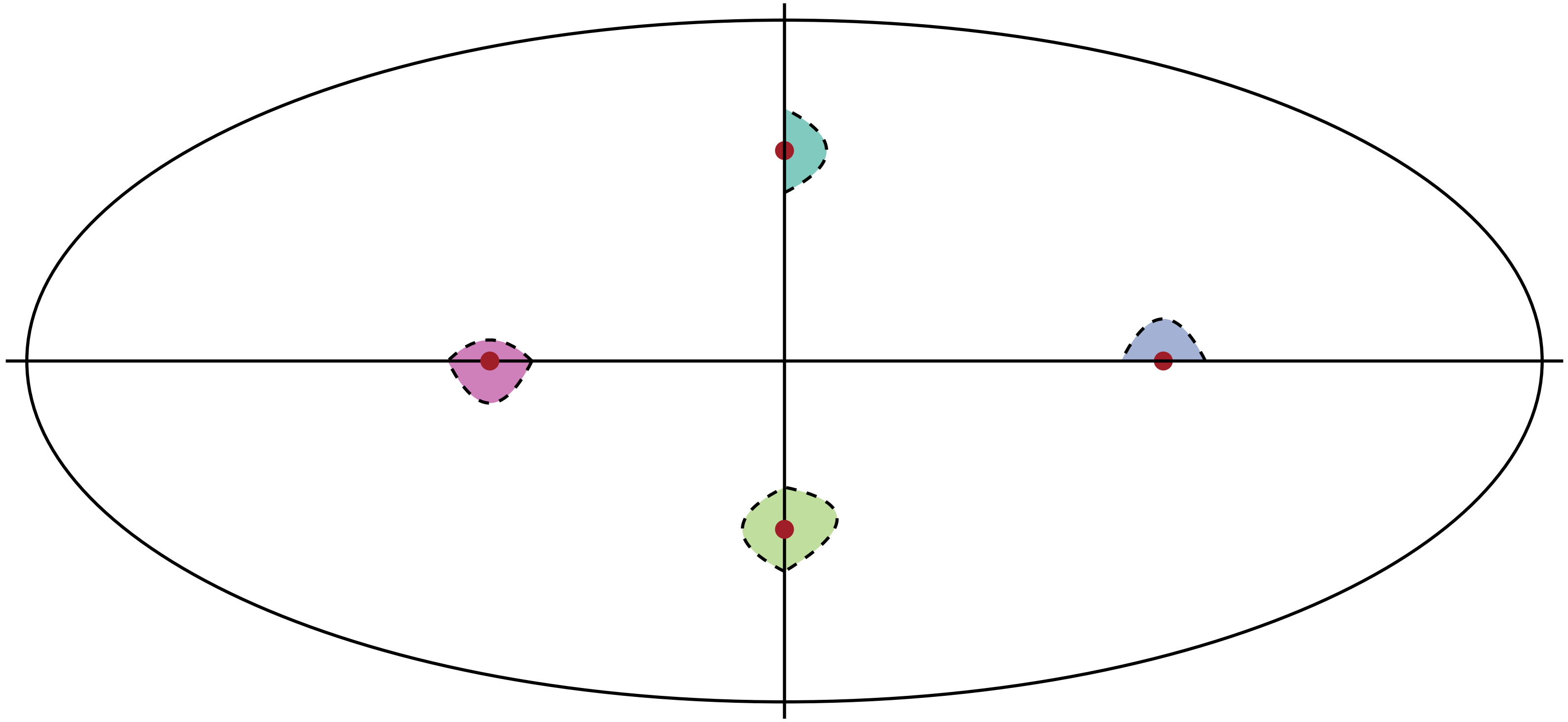
Each thread chooses a random boundary vertex to start



Slides by Y. Akhremtsev

Parallelizing KaHIP - Parallel Local Search

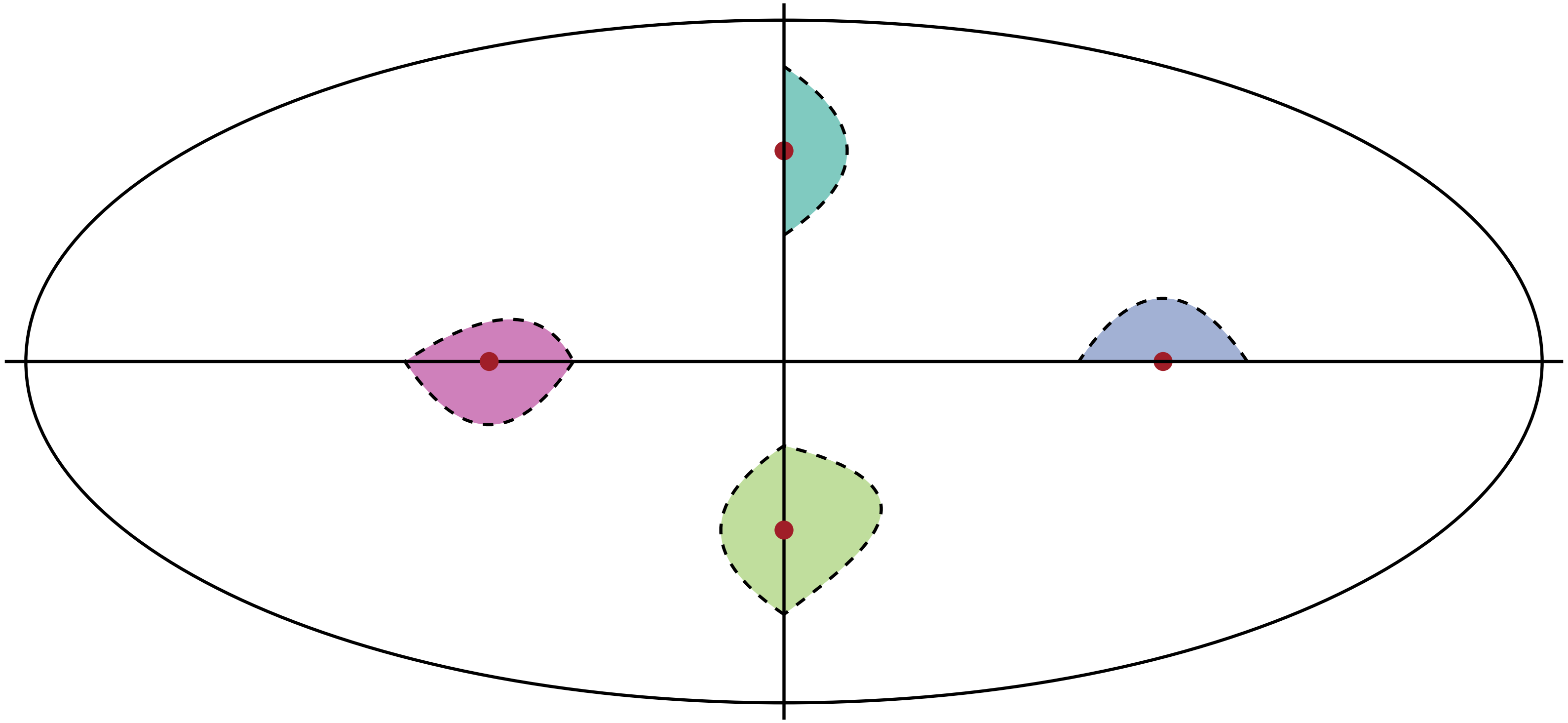
And starts to perform moves around it



Slides by Y. Akhremtsev

Parallelizing KaHIP - Parallel Local Search

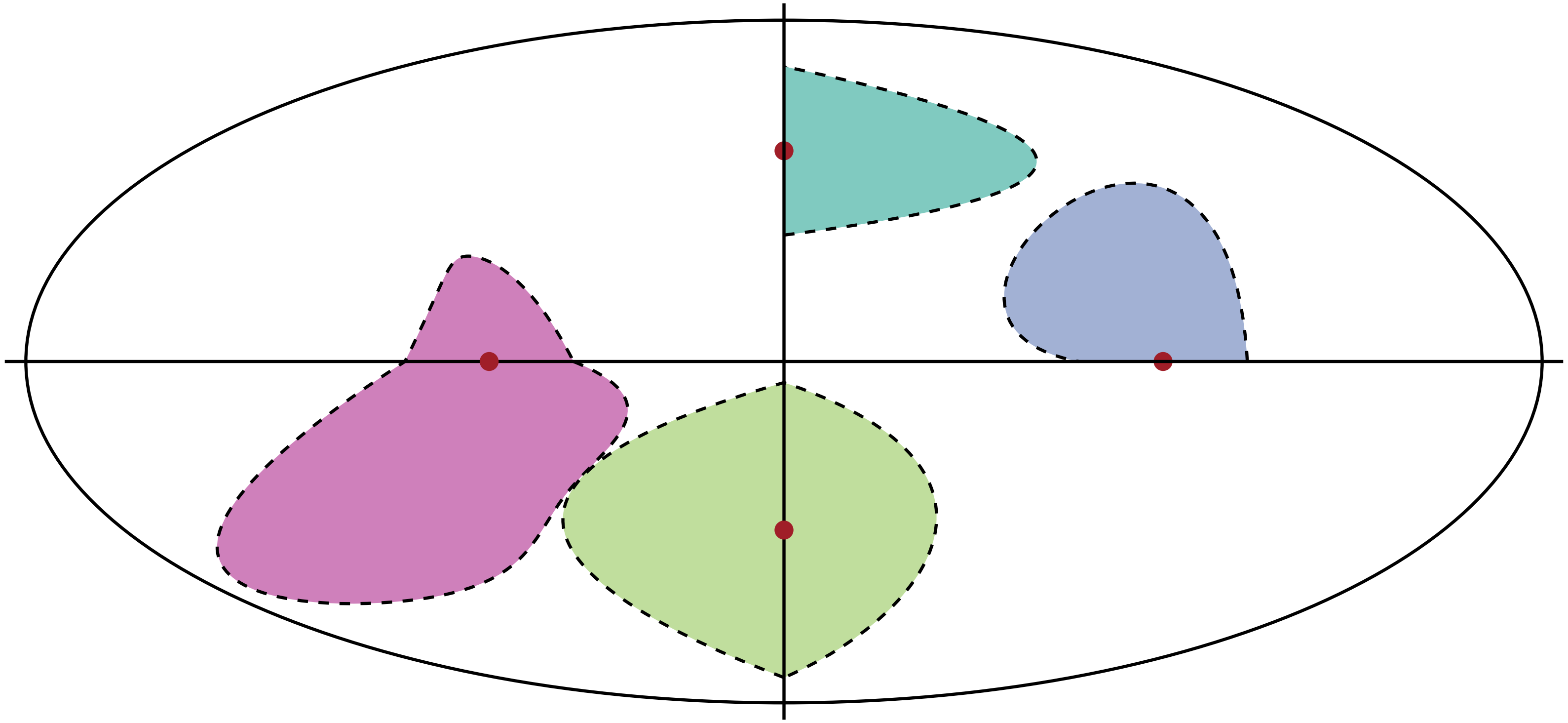
And starts to perform moves around it



Slides by Y. Akhremtsev

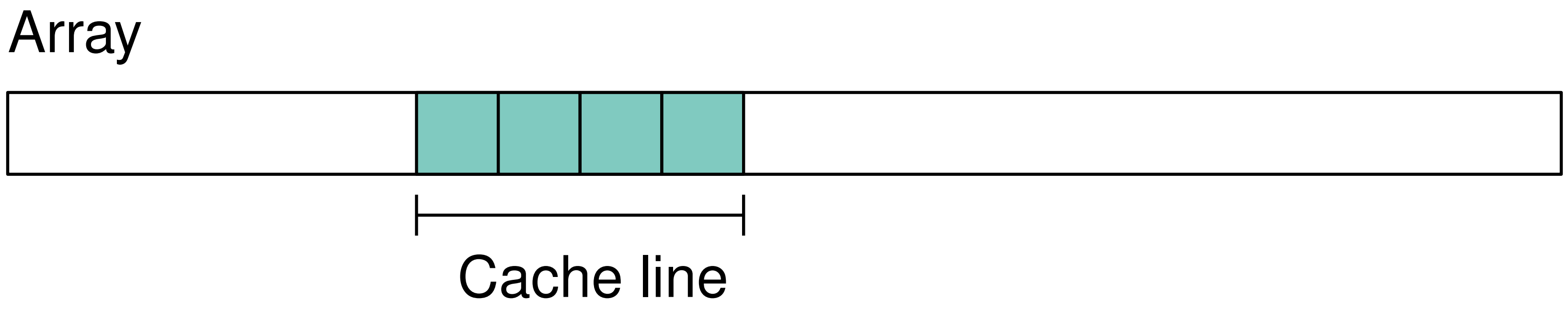
Parallelizing KaHIP - Parallel Local Search

And starts to perform moves around it



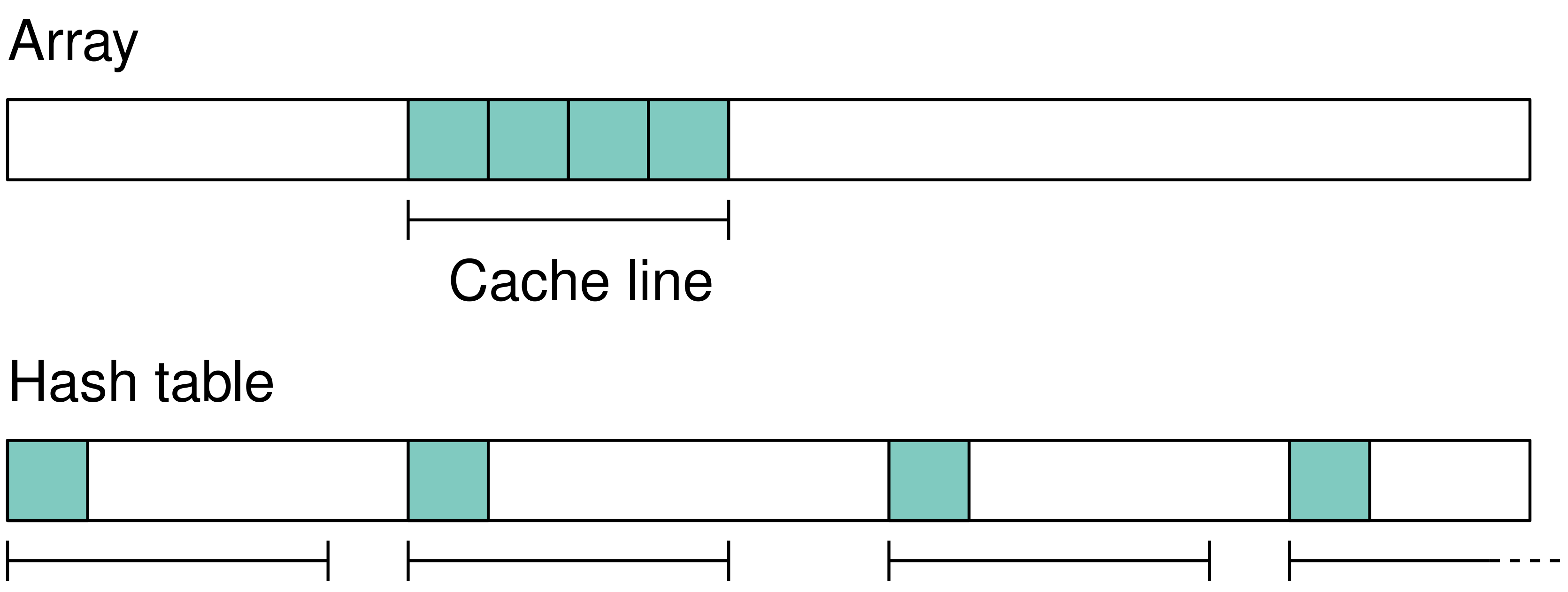
Slides by Y. Akhremtsev

Implementation - Cache-Aware Hash Tables



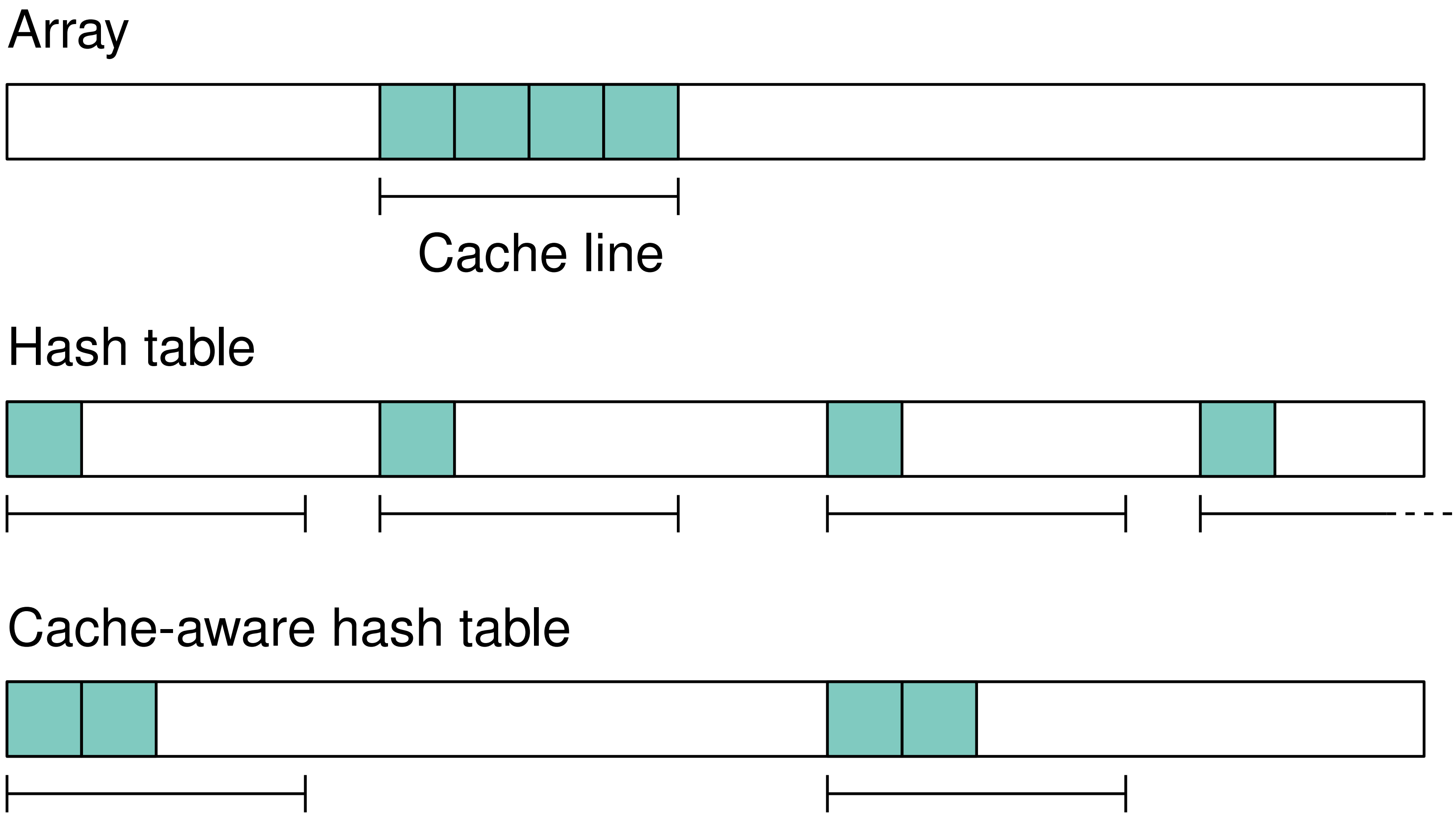
Slides by Y. Akhremtsev

Implementation - Cache-Aware Hash Tables



Slides by Y. Akhremtsev

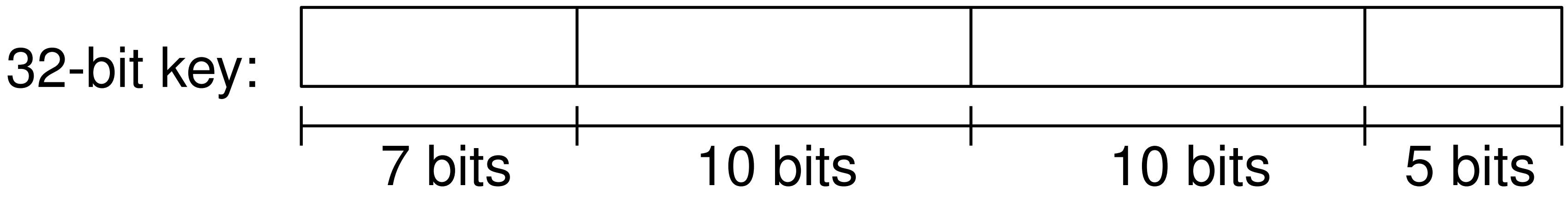
Implementation - Cache-Aware Hash Tables



Slides by Y. Akhremtsev

Implementation - Cache-Aware Hash Tables

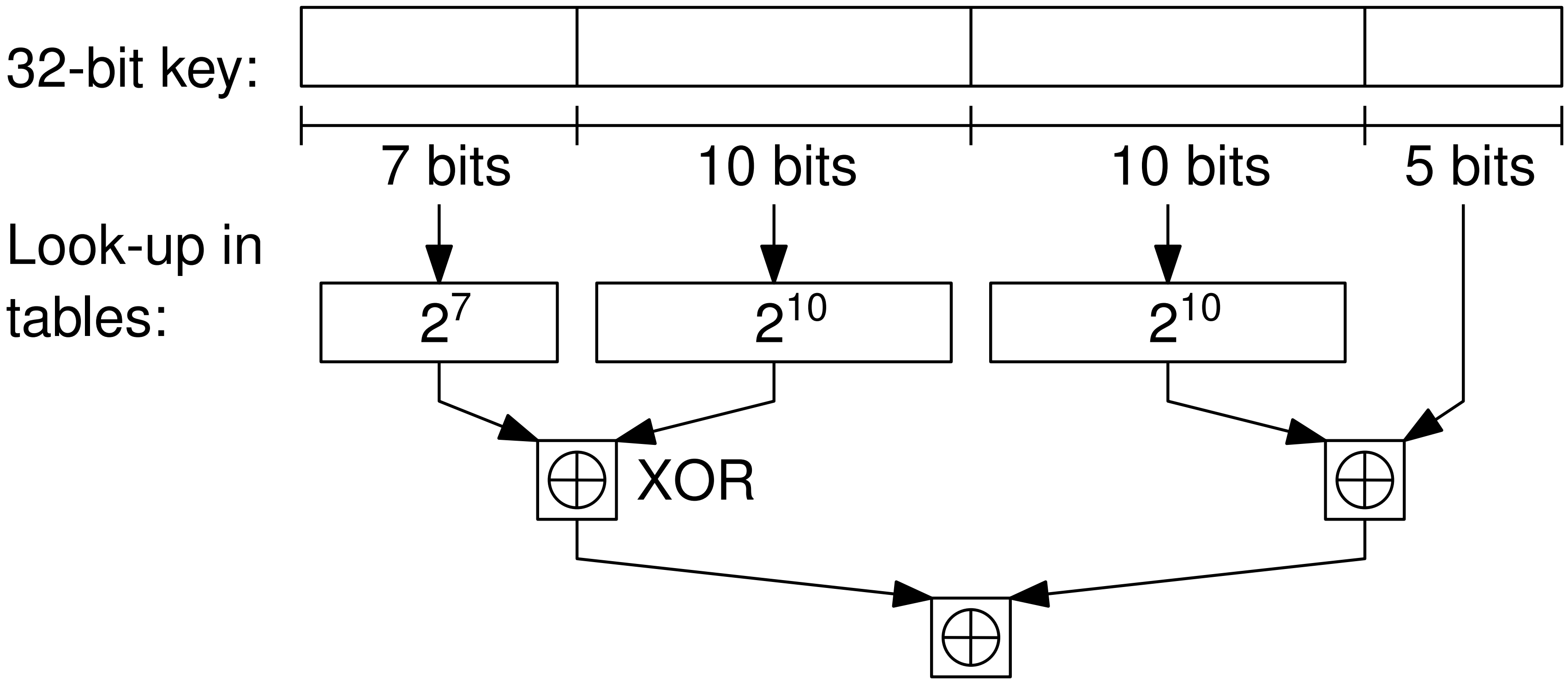
Tabular Hashing



Slides by Y. Akhremtsev

Implementation - Cache-Aware Hash Tables

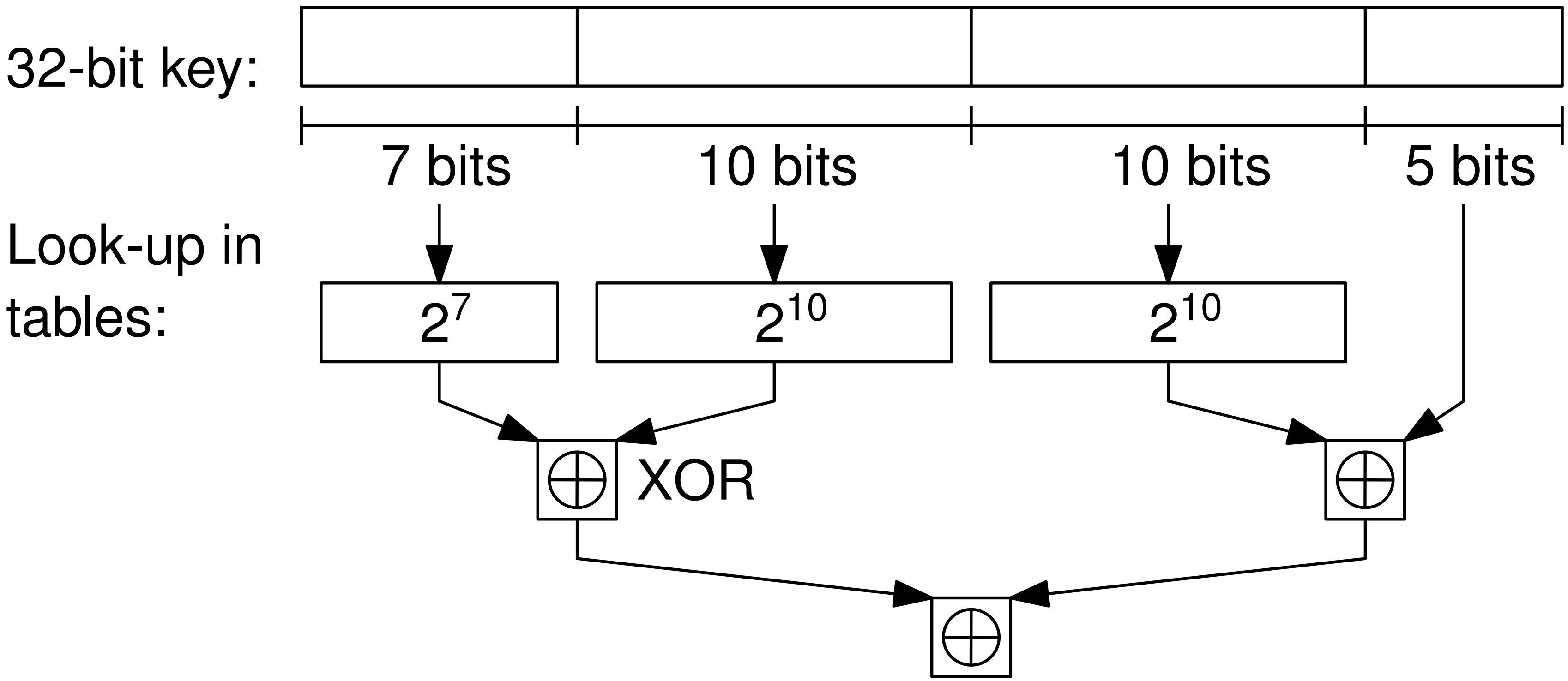
Tabular Hashing



Slides by Y. Akhremtsev

Implementation - Cache-Aware Hash Tables

Tabular Hashing

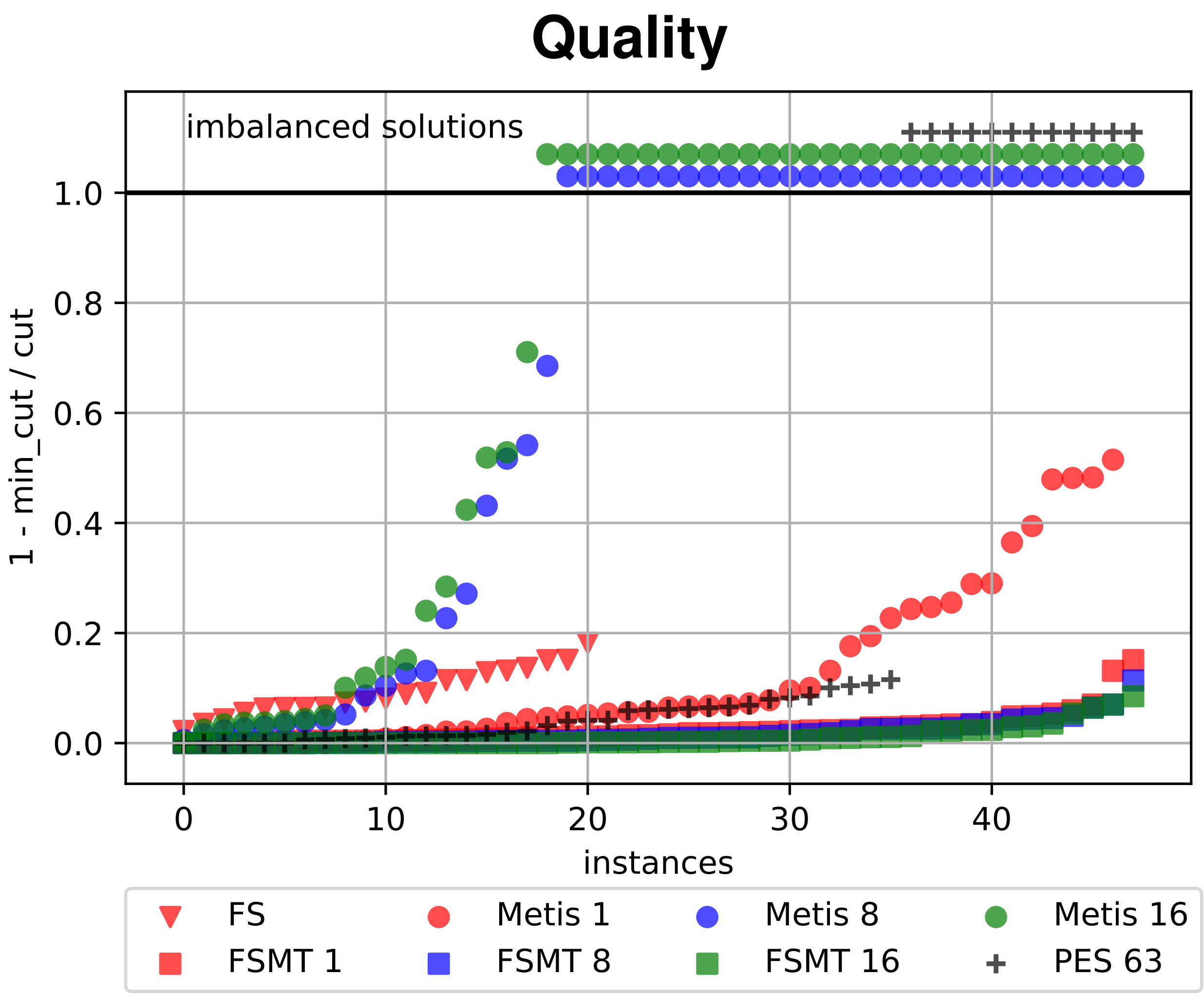


If $|x - y| \leq 2^5$ then $|h(x) - h(y)| \leq 2^5$

Slides by Y. Akhremtsev

Institute of Theoretical Informatics
Algorithmics Group

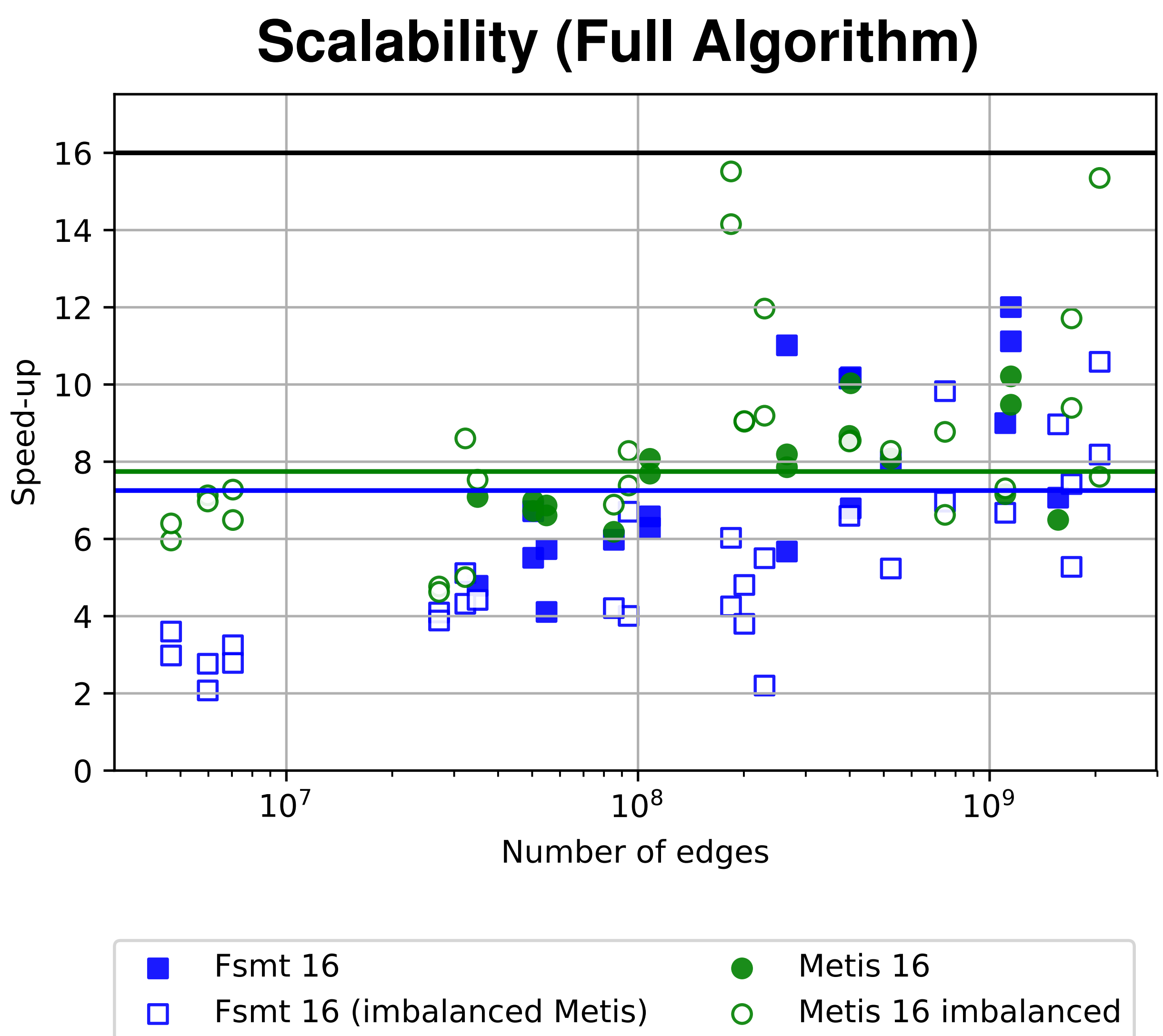
Parallelizing KaHIP - Preliminary Results



Slides by Y. Akhremtsev

Parallelizing KaHIP - Preliminary Results

Karlsruhe Institute of Technology

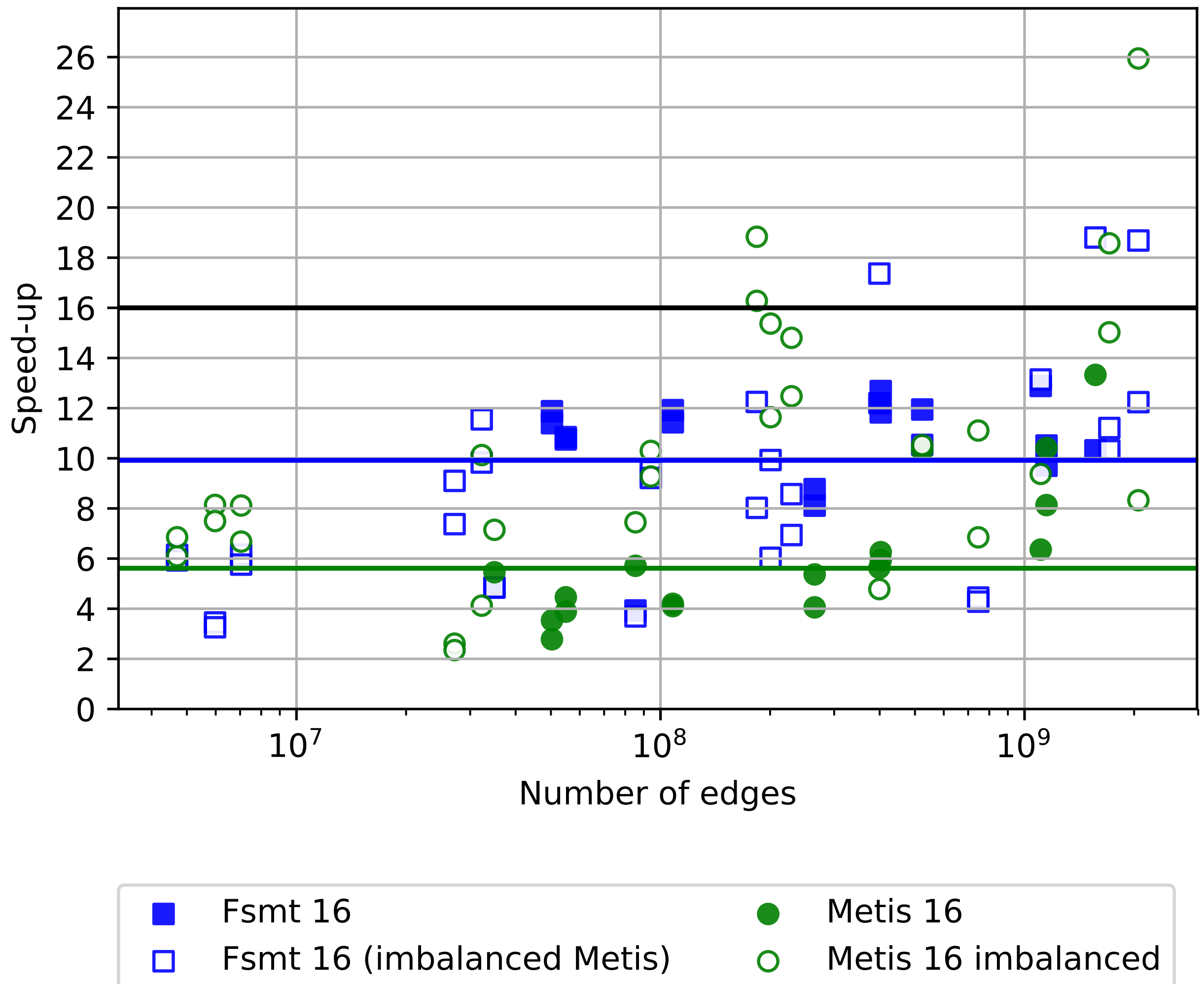


Speed-up for $p = 16$

Slides by Y. Akhremtsev

Parallelizing KaHIP - Preliminary Results

Scalability (Local Search)



Speed-up for p = 16

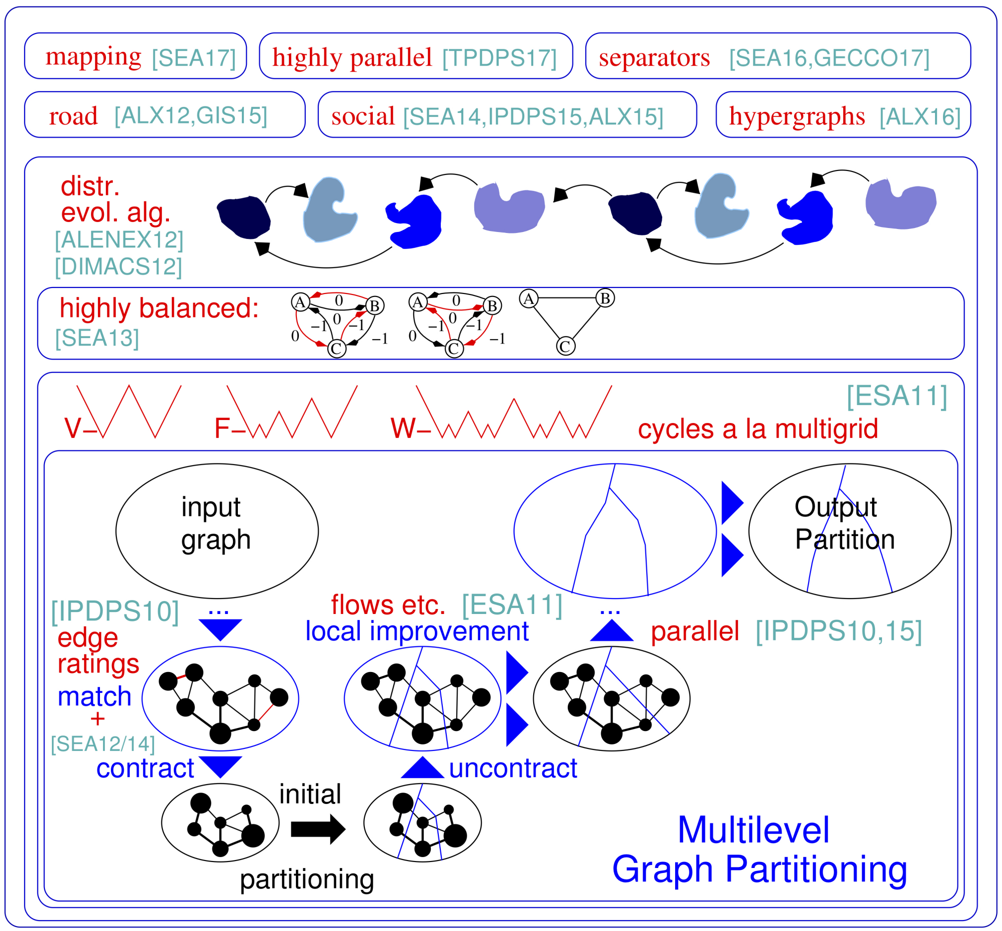
Slides by Y. Akhremtsev

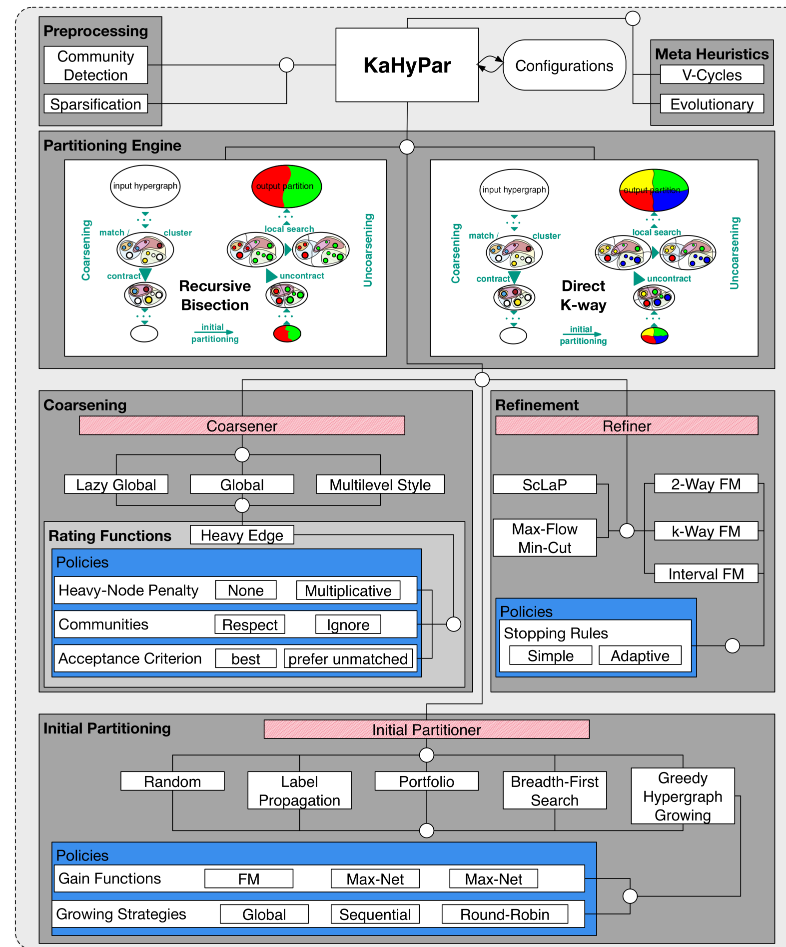
Conclusion

Karlsruhe Institute of Technology

(Hyper)Graph Partitioning Algorithms:

- Graphs: KaHIP – `http://algo2.iti.kit.edu/kahip/`
- Hypergraphs: KaHyPar – `http://www.kahypar.org`





References

Akhremtsev et. al (ALENEX'17): Y. Akhremtsev, T. Heuer, P. Sanders, and S. Schlag. Engineering a direct k-way hypergraph partitioning algorithm. In 19th Workshop on Algorithm Engineering and Experiments, (ALENEX), pages 28–42, 2017.

Heuer, Schlag (SEA'17): T. Heuer and S. Schlag. Improving Coarsening Schemes for Hypergraph Partitioning by Exploiting Community Structure. In 16th International Symposium on Experimental Algorithms, (SEA), page 21:121:19, 2017.

Andre, Schlag, Schulz (arXiv): A., Robin, S. Schlag, and C. Schulz. Memetic Multilevel Hypergraph Partitioning. arXiv preprint arXiv:1710.01968 (2017).

Lamm, Sanders, Schulz (SEA'15): S. Lamm., P. Sanders, C. Schulz. Graph partitioning for independent sets. In 16th International Symposium on Experimental Algorithms, (SEA), page 68-81, 2015.

Lamm et. al (ALENEX'16): Lamm, S., Sanders, P., Schulz, C., Strash, D., Werneck, R. F. (2017). Finding near-optimal independent sets at scale. Journal of Heuristics, 23(4), 207-229.

Ahuja et. al (GIS'15): Ahuja, N., Bender, M., Sanders, P., Schulz, C., Wagner, A. (2015, November). Incorporating road networks into territory design. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems (p. 4). ACM.

Sanders, Schulz (SEA'16): Sanders, P., Schulz, C. Advanced Multilevel Node Separator Algorithms. In 16th International Symposium on Experimental Algorithms, (SEA), pages 294-309, 2016.

Schulz et. al (GECCO'17): P. Sanders, C. Schulz, D. Strash, R. Williger. 2017. Distributed evolutionary k-way node separators. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '17).

Schulz, Träff (SEA'17): C. Schulz and J. Larsson Träff. Better Process Mapping and Sparse Quadratic Assignment. In 16th International Symposium on Experimental Algorithms, (SEA), page 4:1–4:15 2017.

Hamann, Strasser (ALENEX'16): Hamann, M., Strasser, B. (2016). Graph bisection with pareto-optimization. I. In 19th Workshop on Algorithm Engineering and Experiments, (ALENEX), pages 90–102, 2017.

Moreia, Popp, Schulz (SEA'17): O. Moreira, M. Popp, C. Schulz. Graph Partitioning with Acyclicity Constraints. In 16th International Symposium on Experimental Algorithms, (SEA), pages 30:1–30:15, 2017.