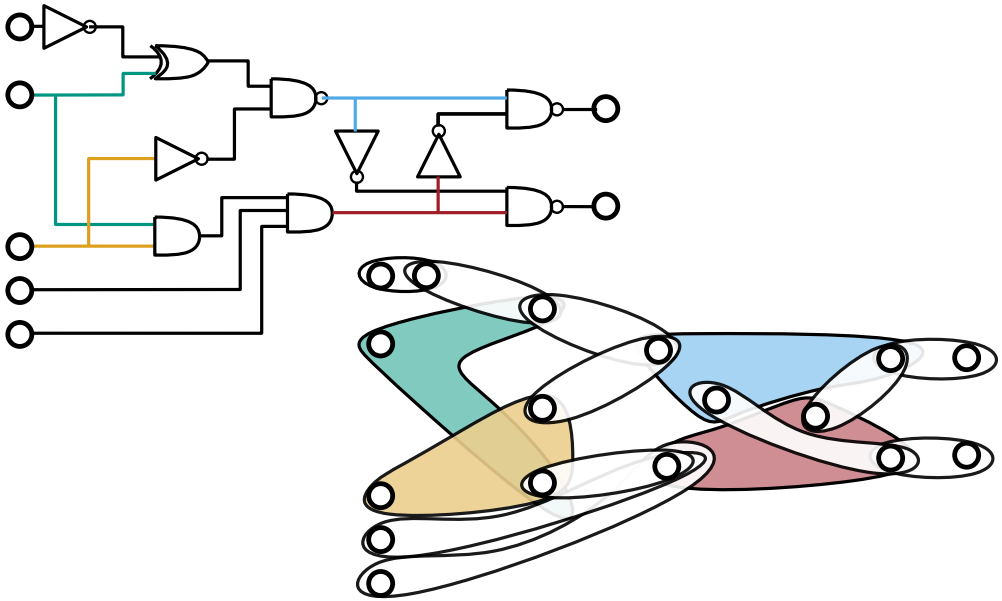
Karlsruhe Institute of Technology

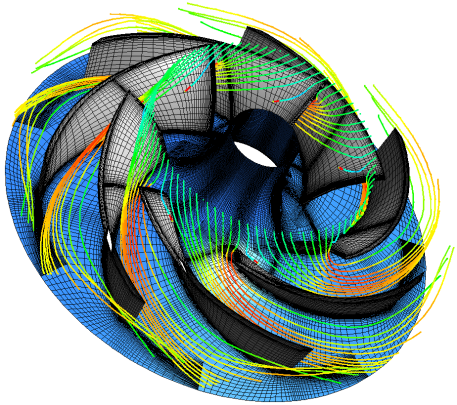
News on (Hyper)graph Partitioning

Annual SPP Meeting · September 25, 2018

Sebastian Schlag

INSTITUTE OF THEORETICAL INFORMATICS ·





ε-balanced (Hyper)graph Partitioning

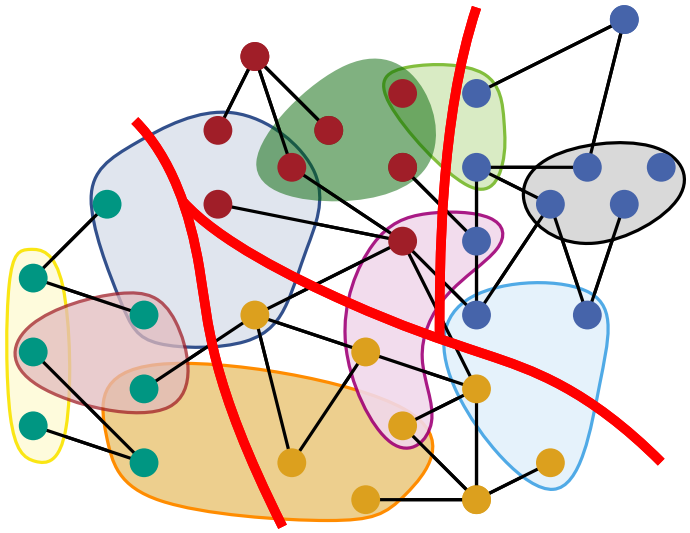
Partition (hyper)graph $G = (V, E, c : V \to \mathbf{R}_{>0}, \omega : E \to \mathbf{R}_{>0})$ into k disjoint blocks s.t.

- node weight of each block $\leq \frac{1+\varepsilon}{k}$ total node weight
- total weight of cut (hyper)edges as small as possible
- or connectivity, i.e. $\sum_{e \in \text{cut}} (\lambda - 1)\omega(e)$

blocks of hyperedge e

Applications:

- VLSI Design
- Scientific Computing
- Route Planning
- ...



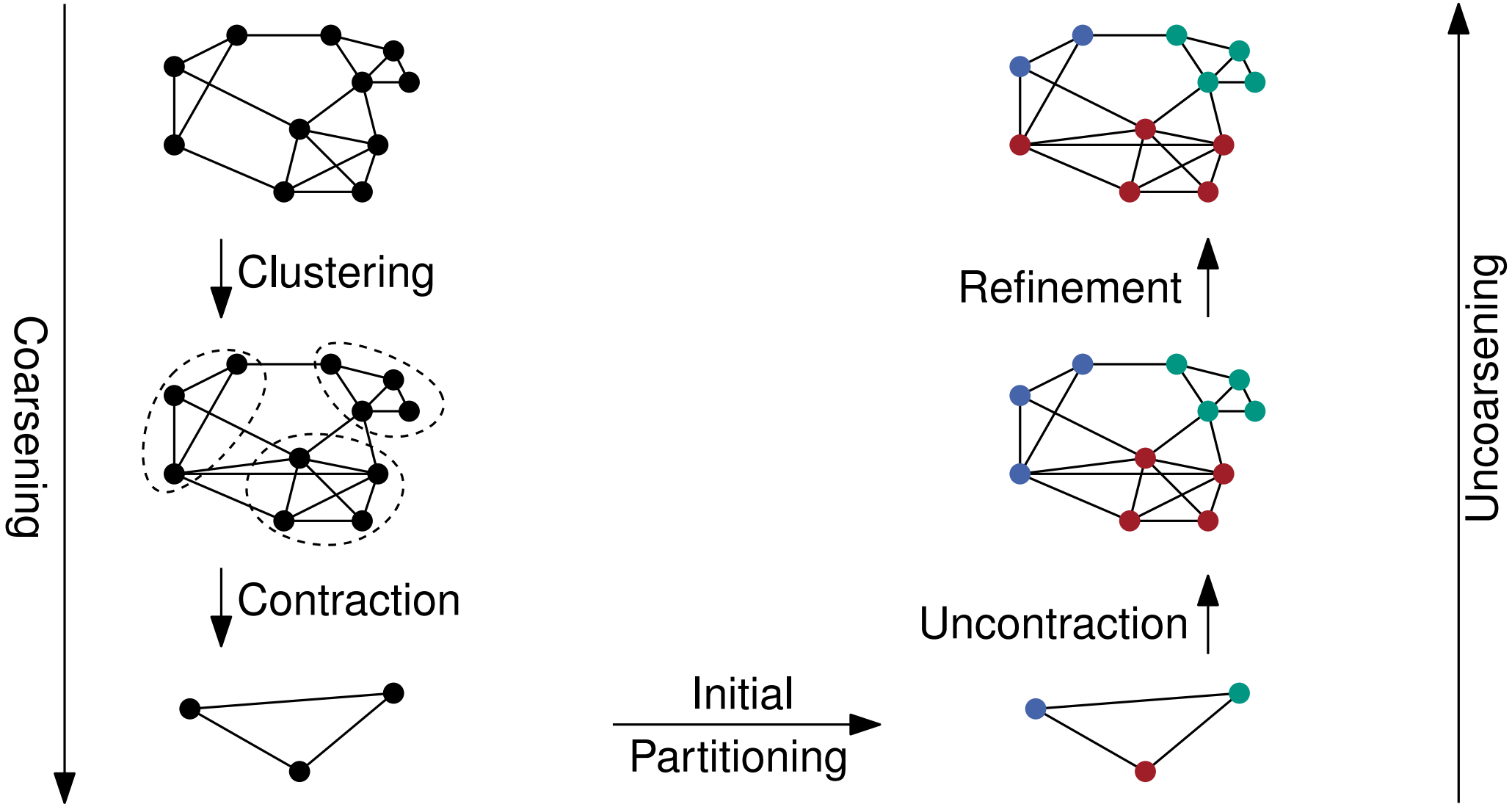
News on (Hyper)graph Partitioning

- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

News on (Hyper)graph Partitioning

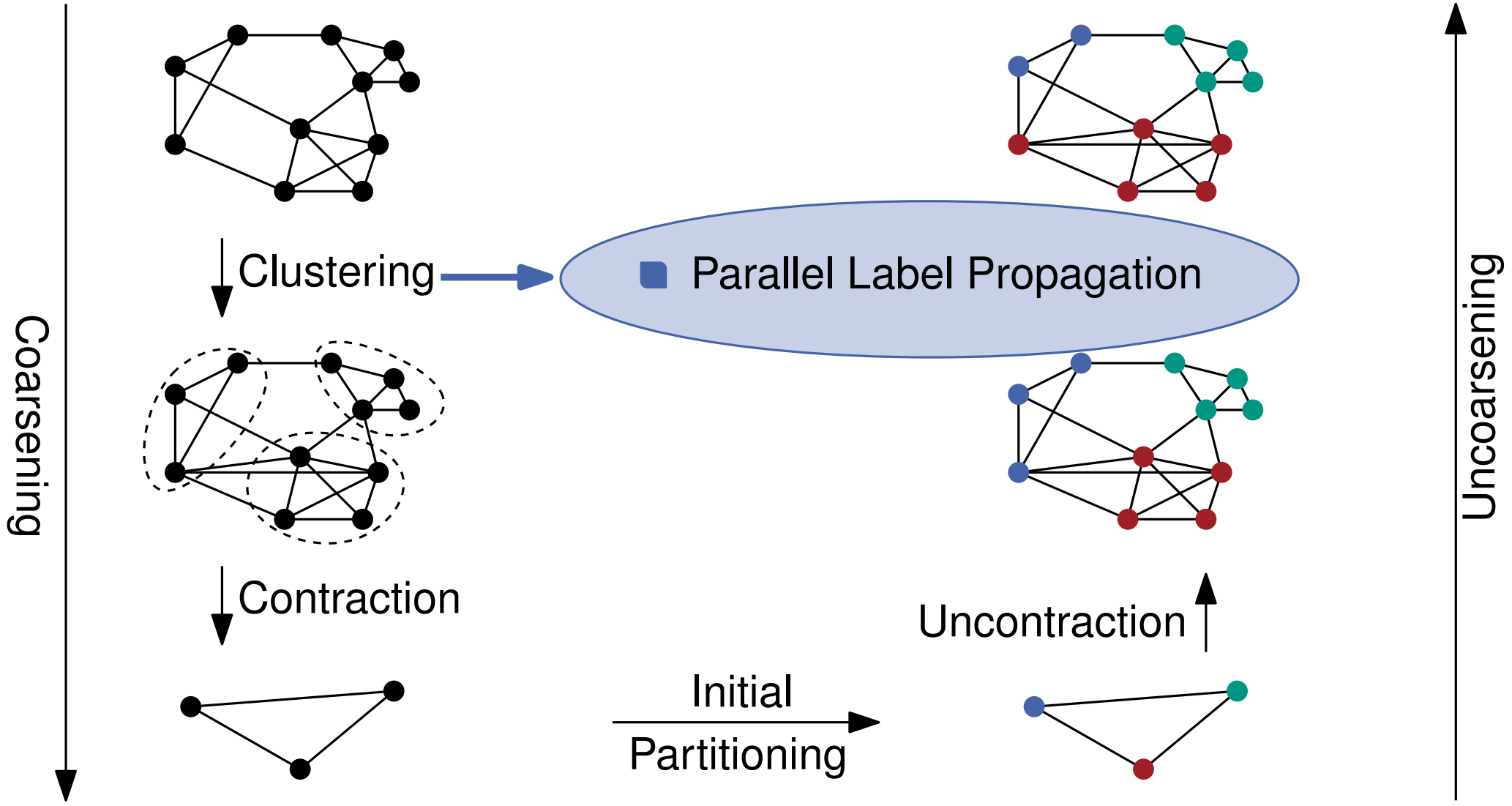
- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

High-Quality Shared Memory Graph Partitioning



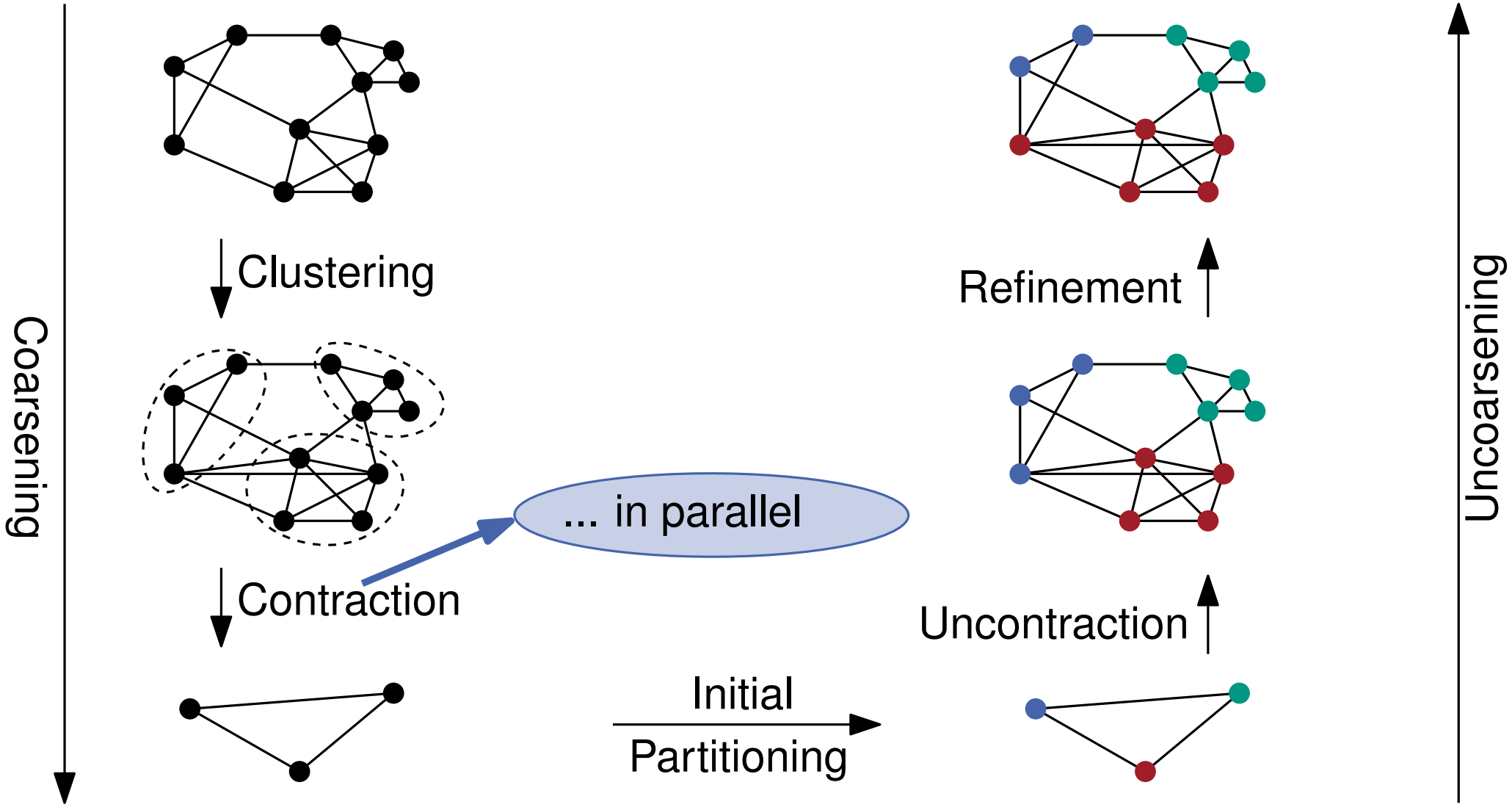
[Y. Akhremtsev]

High-Quality Shared Memory Graph Partitioning



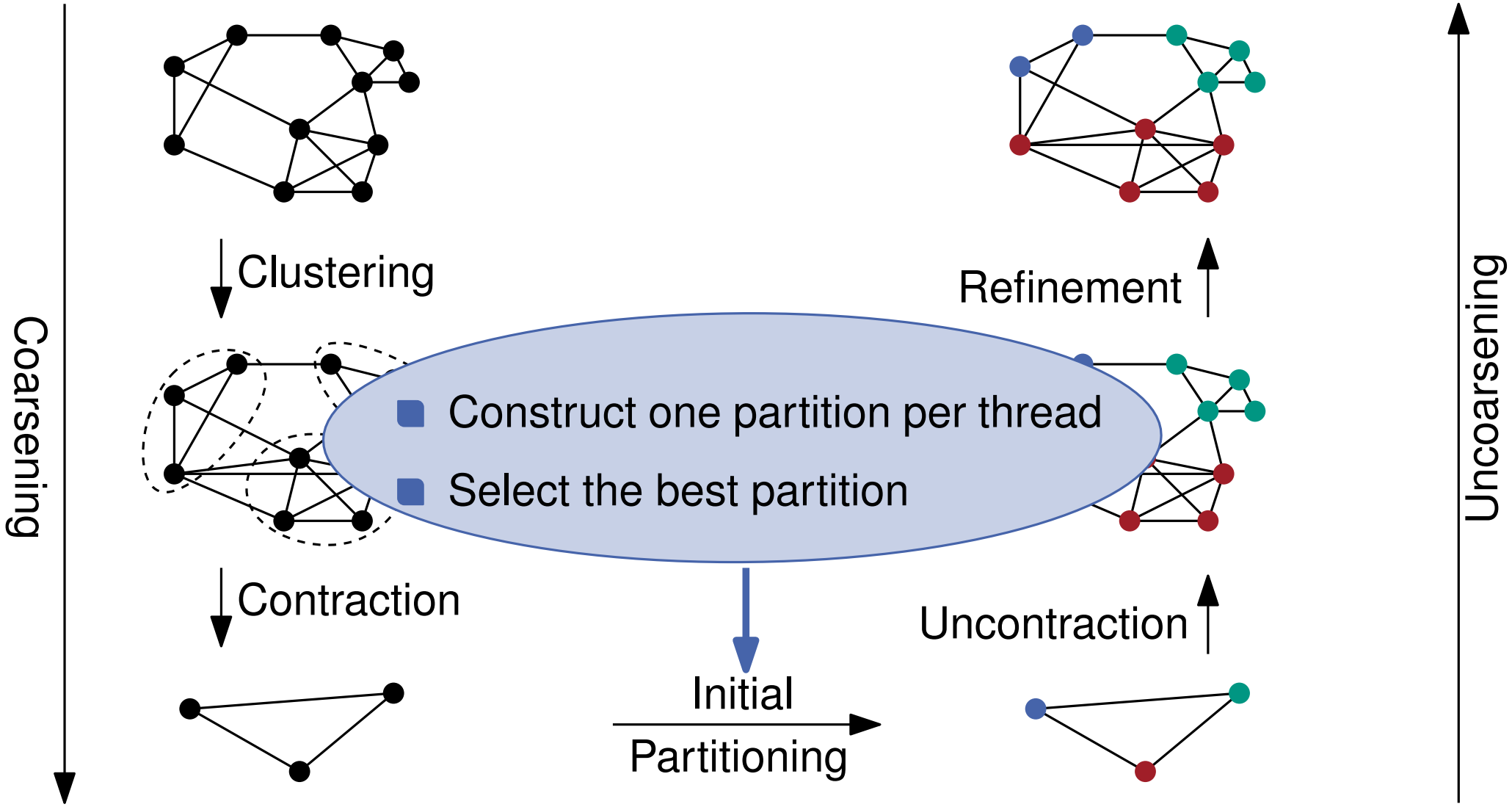
[Y. Akhremtsev]

High-Quality Shared Memory Graph Partitioning



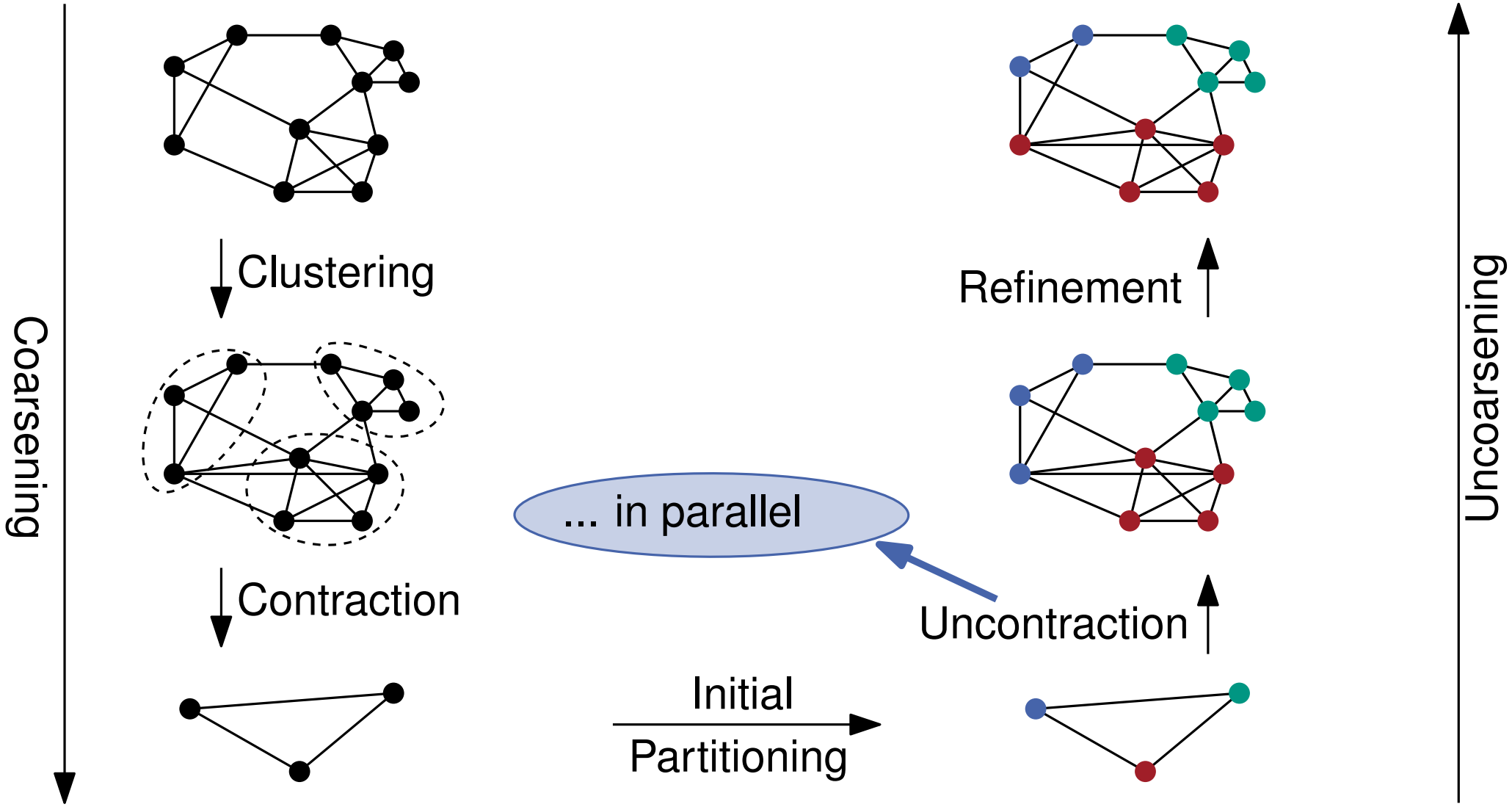
[Y. Akhremtsev]

High-Quality Shared Memory Graph Partitioning



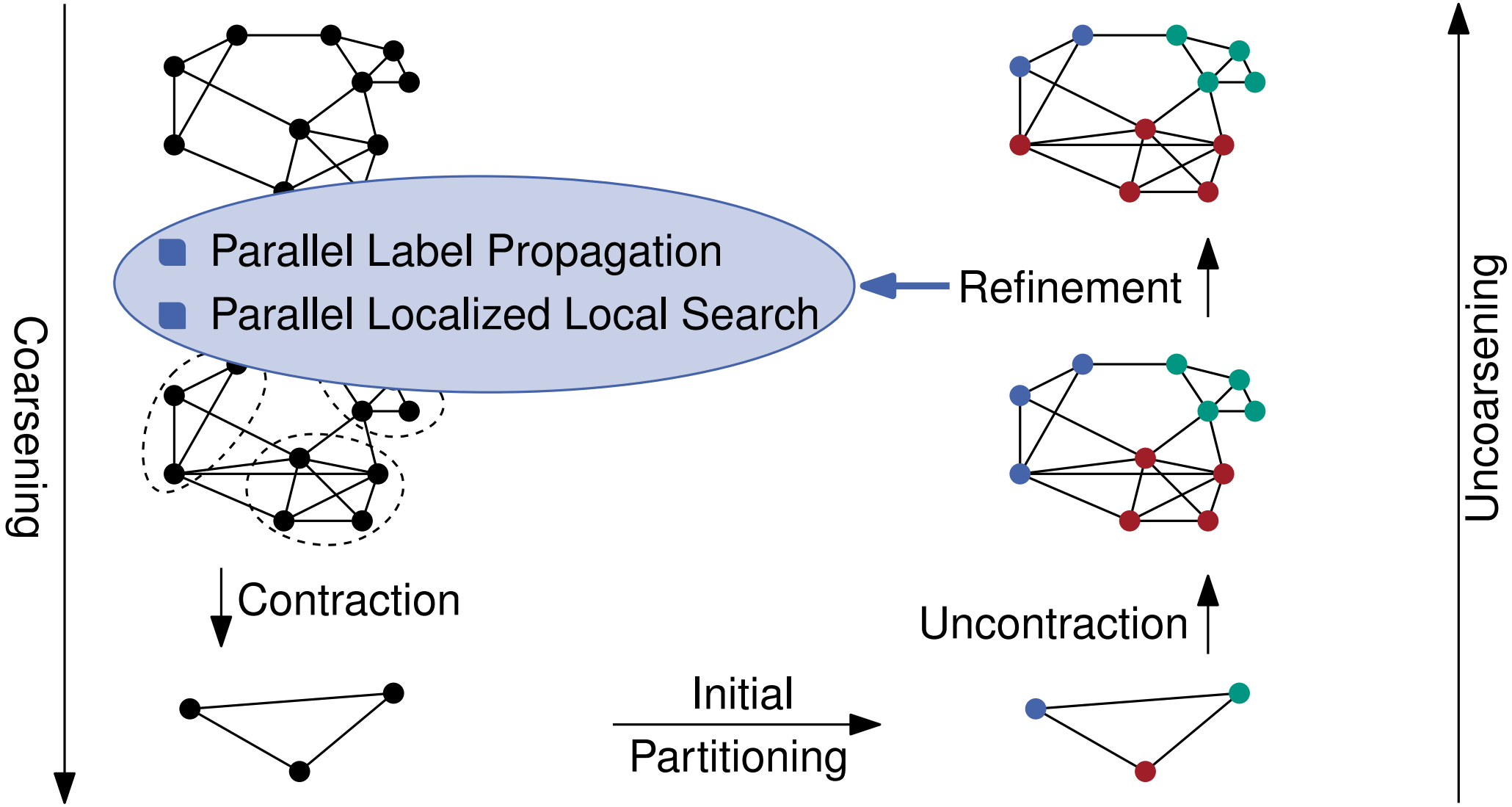
[Y. Akhremtsev]

High-Quality Shared Memory Graph Partitioning



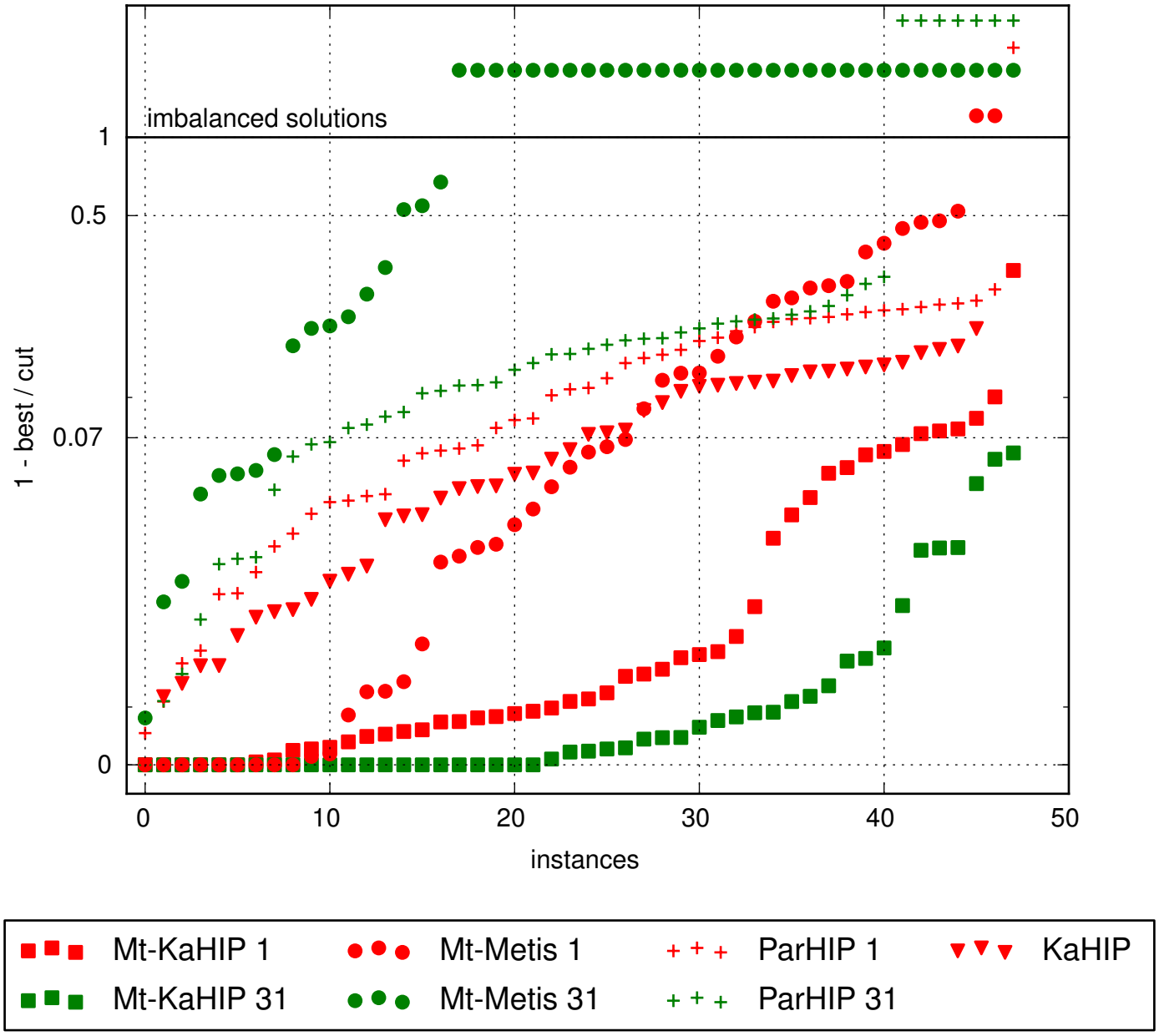
[Y. Akhremtsev]

High-Quality Shared Memory Graph Partitioning



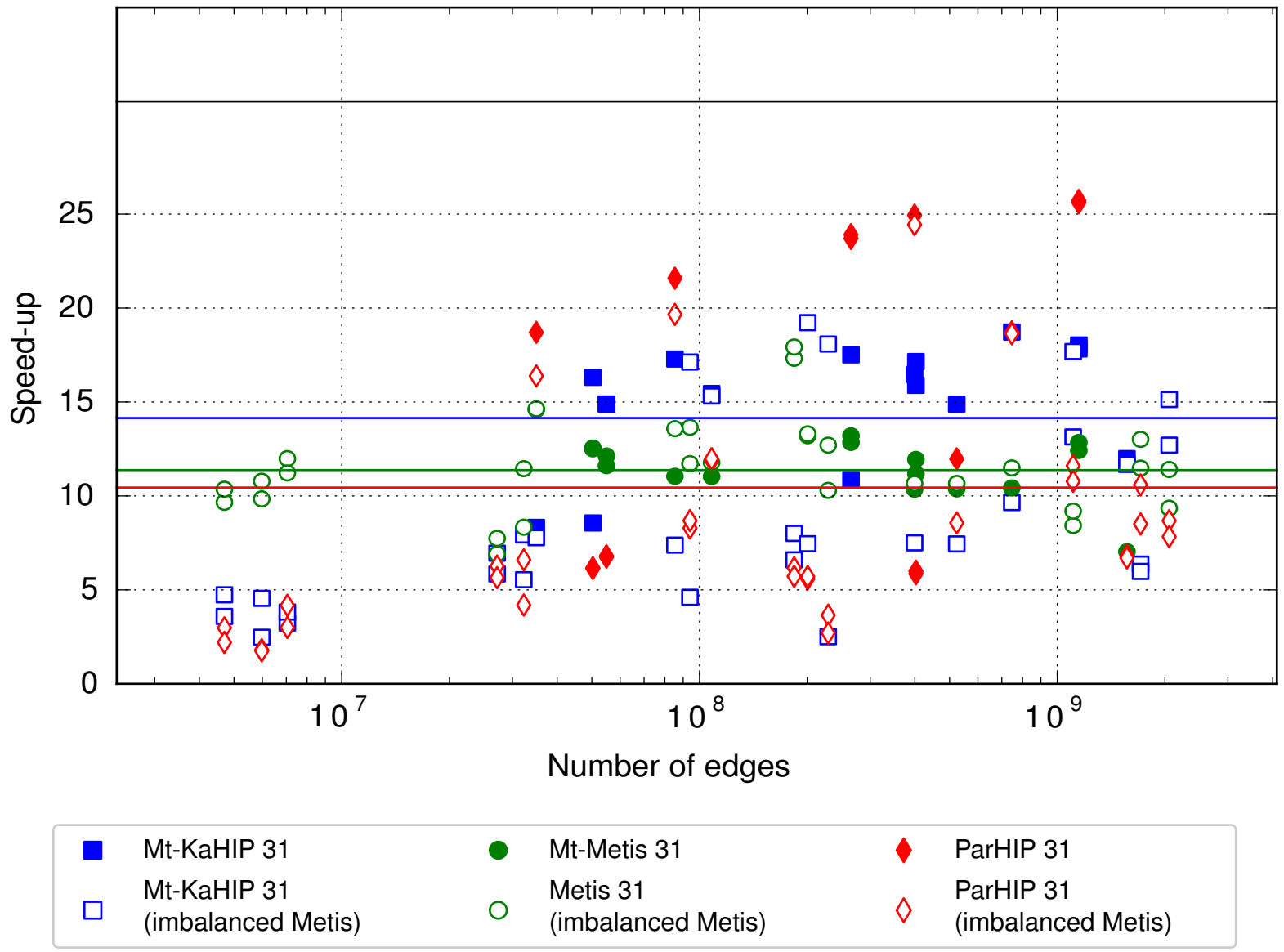
[Y. Akhremtsev]

Mt-KaHIP – Solution Quality



[Y. Akhremtsev]

Mt-KaHIP – Total Speed-Up



[Y. Akhremtsev]

News on (Hyper)graph Partitioning

- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

News on (Hyper)graph Partitioning

- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

GP – Integer Linear Program

$$\min \sum_{\{u,v\} \in E} e_{uv} \cdot \omega(\{u,v\}) \tag{1}$$

$$\forall \{u,v\} \in E, \forall k : e_{uv} \geq |x_{u,k} - x_{v,k}| \tag{2}$$

$$\forall k : \sum_{v \in V} x_{v,k} c(v) \leq L_{\max} \tag{3}$$

$$\forall v \in V : \sum_{k} x_{v,k} = 1 \tag{4}$$

Variables:

- binary decision variables for all edges and vertices
- for $\{u,v\} \in E$ use $e_{uv} \in \{0,1\}$: one **iff** cut edge
- for $v \in V$ and block k use $x_{v,k} \in \{0,1\}$: one **iff** v is in block k

[C. Schulz]

GP – Integer Linear Program

$$\min \sum_{\{u,v\} \in E} e_{uv} \cdot \omega(\{u,v\}) \tag{1}$$

$$\forall \{u,v\} \in E, \forall k : e_{uv} \geq |x_{u,k} - x_{v,k}| \tag{2}$$

$$\forall k : \sum_{v \in V} x_{v,k} c(v) \leq L_{\max} \tag{3}$$

$$\forall v \in V : \sum_{k} x_{v,k} = 1 \tag{4}$$

Variables:

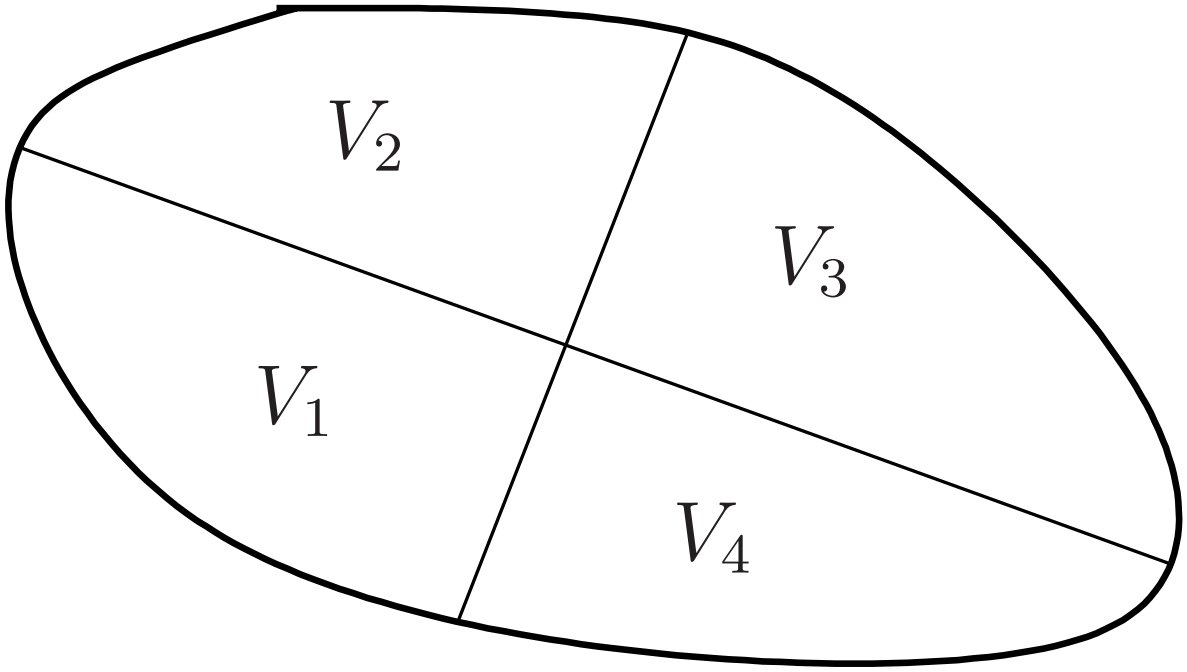
- binary decision variables for all edges and vertices
- for $\{u,v\} \in E$ use $e_{uv} \in \{0,1\}$: one **iff** cut edge
- for $v \in V$ and block k use $x_{v,k} \in \{0,1\}$: one **iff** v is in block k

ILP does NOT scale to large graphs

[C. Schulz]

Local Search using ILP

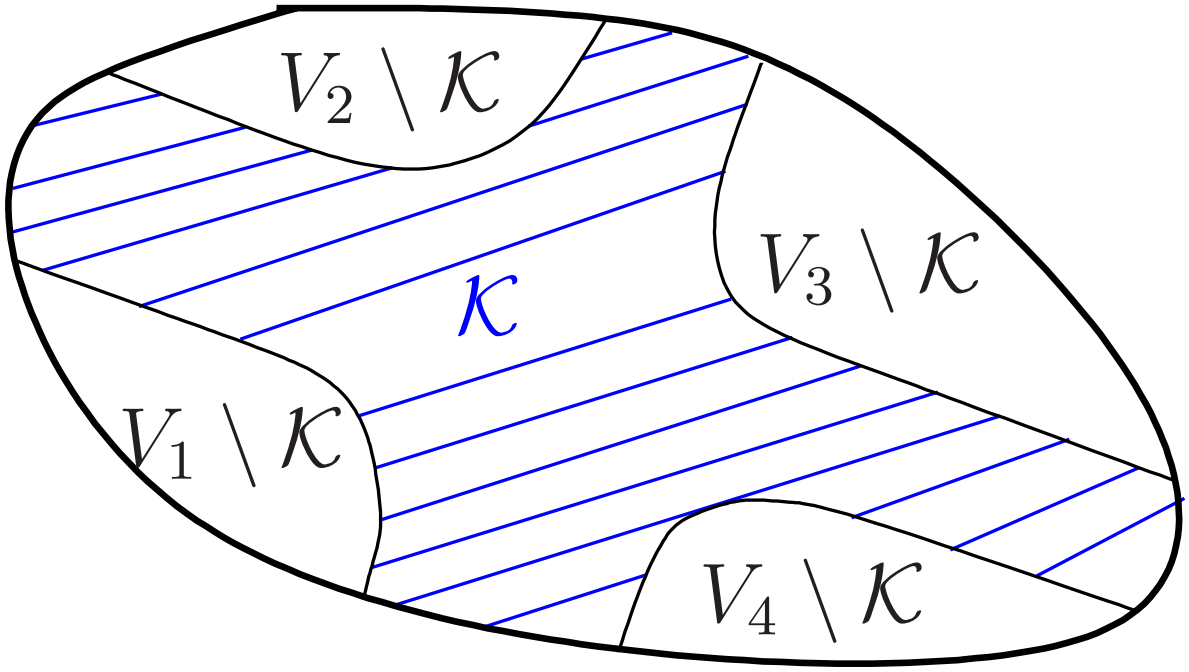
- use given partition as input
- contract areas far away from boundary
- solve weighted problem using ILP



[C. Schulz]

Local Search using ILP

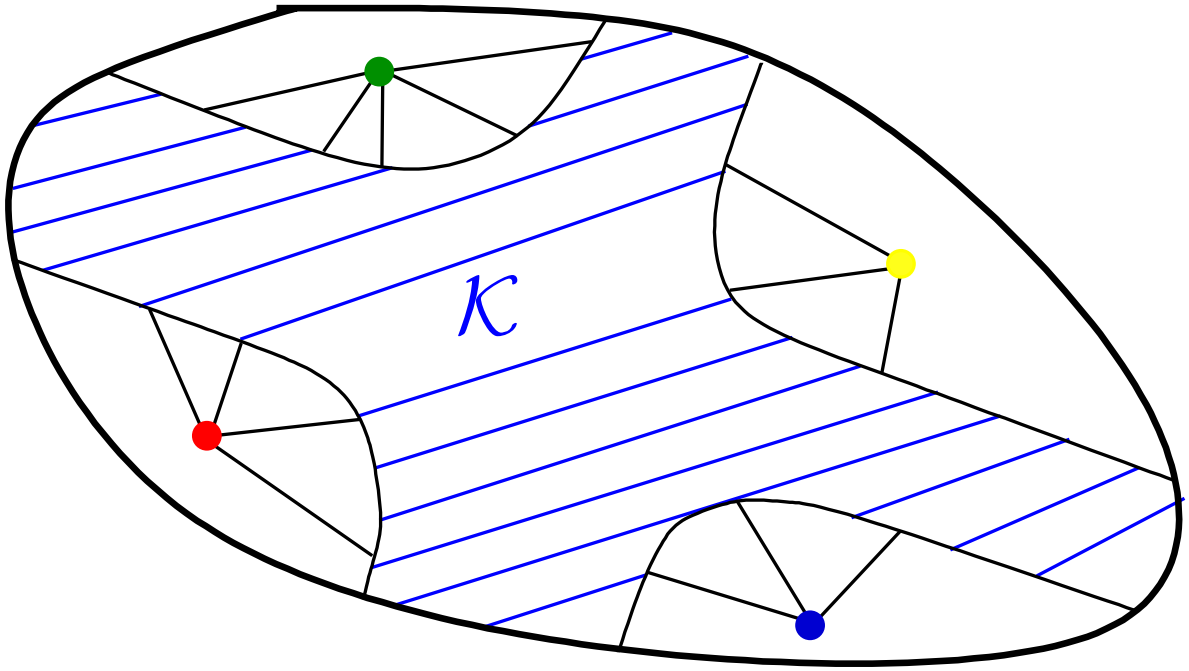
- use given partition as input
- contract areas far away from boundary
- solve weighted problem using ILP



[C. Schulz]

Local Search using ILP

- use given partition as input
- contract areas far away from boundary
- solve weighted problem using ILP



Other optimizations:

symmetry breaking, use start solution, solver parallelism, ...

[C. Schulz]

Walshaw Benchmark

Number of records, using current records as input

k/ϵ	0%	1%	3%	5%
2	6%	12%	6%	6%
4	18%	9%	6%	18%
8	26%	24%	12%	15%
16	50%	26%	29%	29%
32	62%	47%	47%	53%
64	68%	59%	71%	76%
sum	38%	29%	28%	33%

Σ 261 better partitions

[C. Schulz]

News on (Hyper)graph Partitioning

- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

News on (Hyper)graph Partitioning

- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

Evolutionary Algorithms for HGP

	#	Population	Recombination	Mutation	LS	ML
Saab & Rao '89	k	bin packing	✗	rand. greedy	✗	–
Hulin '91	2	rand.	2-point	rand.	✗	–
Bui & Moon '94	2	rand.	5-point	rand.	FM	–
Areibi '00	k	rand.	3/4-point	rand.	kFM	–
Areibi & Yang '04	k	rand./GRASP	3/4-point	rand.	kFM	–
Kim et al. '04	2	rand.	5-point	rebalance	FM	–
Armstrong et al. '10	k	rand./kFM	2-point	rand.	kFM	–

- **no** usage of multilevel paradigm
- considered **not** competitive with state-of-the-art tools [Cohoon et al. '03]
- benchmarked on **small** & **outdated** hypergraphs

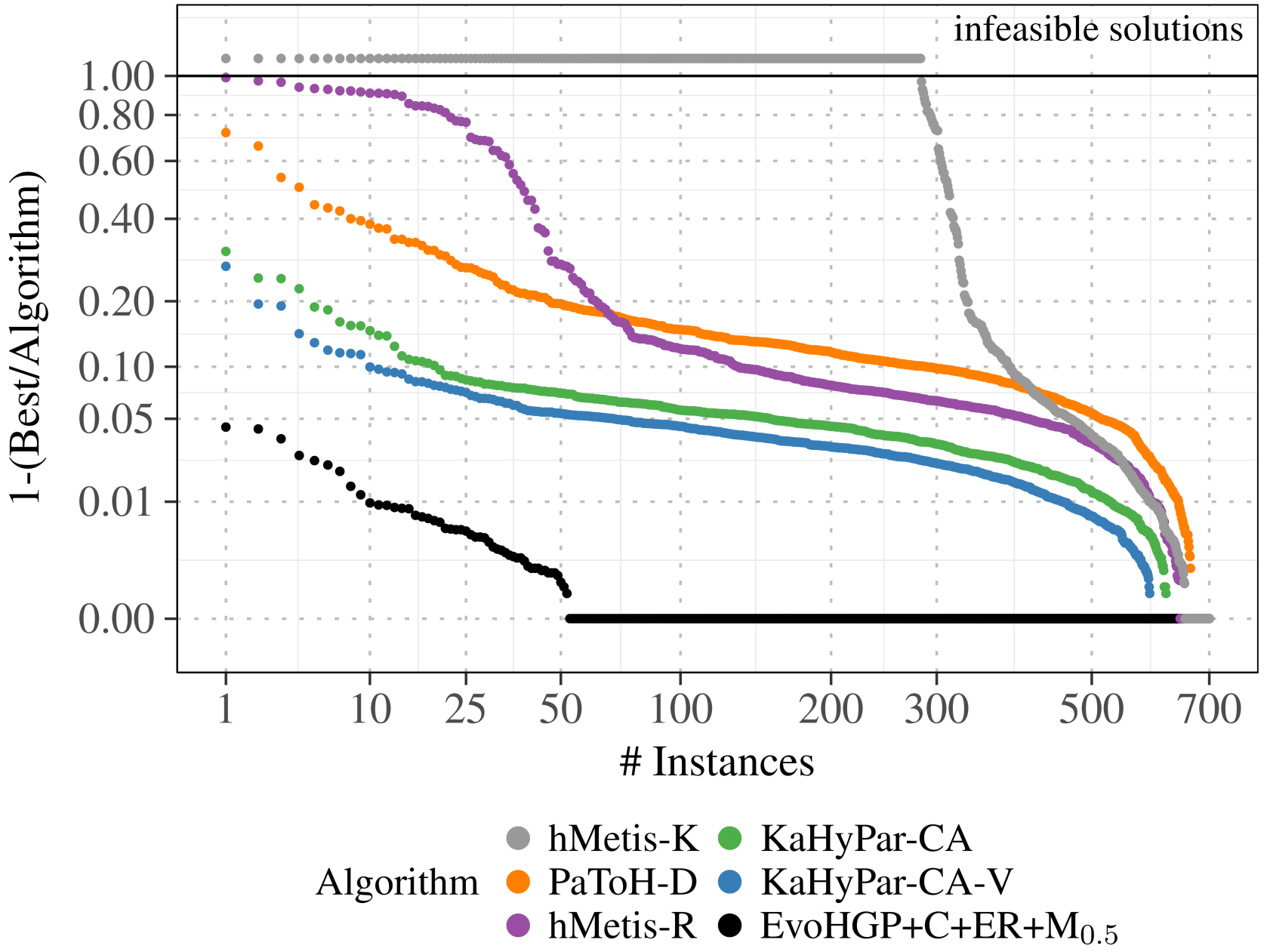
Evolutionary Algorithms for HGP & Contributions

	#	Population	Recombination	Mutation	LS	ML
Saab & Rao '89	k	bin packing	✗	rand. greedy	✗	–
Hulin '91	2	rand.	2-point	rand.	✗	–
Bui & Moon '94	2	rand.	5-point	rand.	FM	–
Areibi '00	k	rand.	3/4-point	rand.	kFM	–
Areibi & Yang '04	k	rand./GRASP	3/4-point	rand.	kFM	–
Kim et al.					FM	–
Armstrong					kFM	–

Our Contribution:
first **memetic multilevel** HGP algorithm
$\Rightarrow$ problem-**specific** recombine & mutation operators
$\Rightarrow$ **extensive** experiments on **large** benchmark set

- **no** usage of multilevel paradigm
- considered **not** competitive with state-of-the-art tools [Cohoon et al. '03]
- benchmarked on **small** & **outdated** hypergraphs

Comparison with Best Non-Evo Algorithms



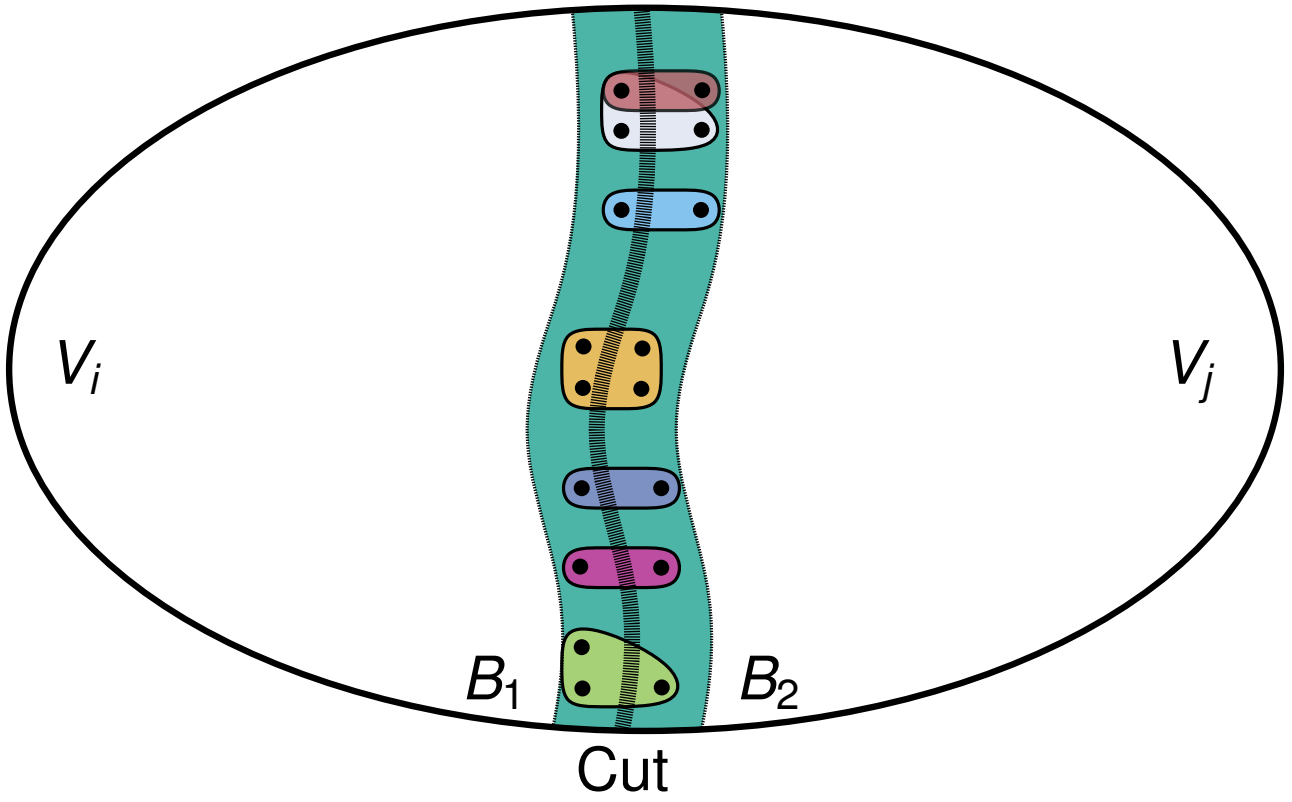
News on (Hyper)graph Partitioning

- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

News on (Hyper)graph Partitioning

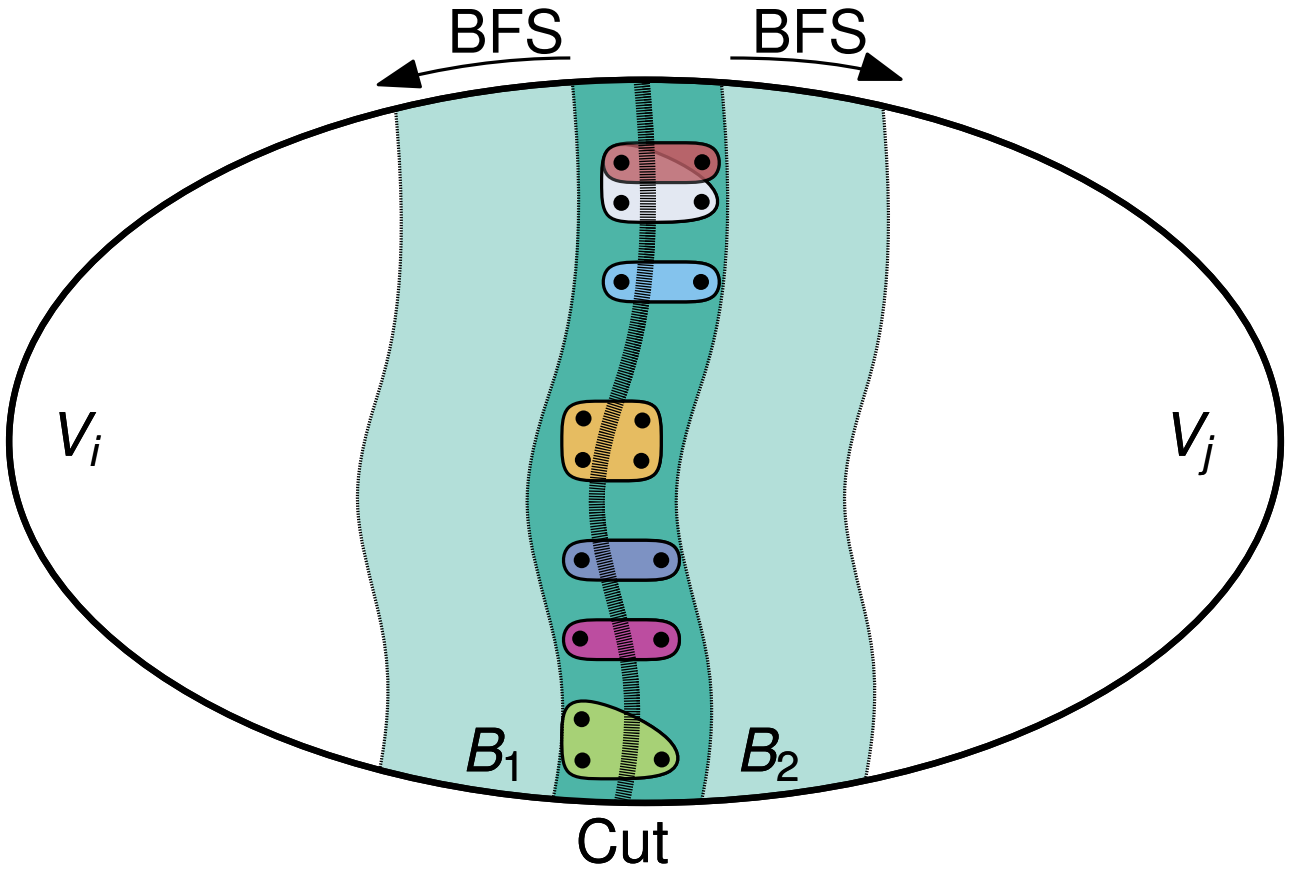
- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

KaFFPa's Flow-Based Refinement for hypergraphs



KaFFPa's Flow-Based Refinement for hypergraphs

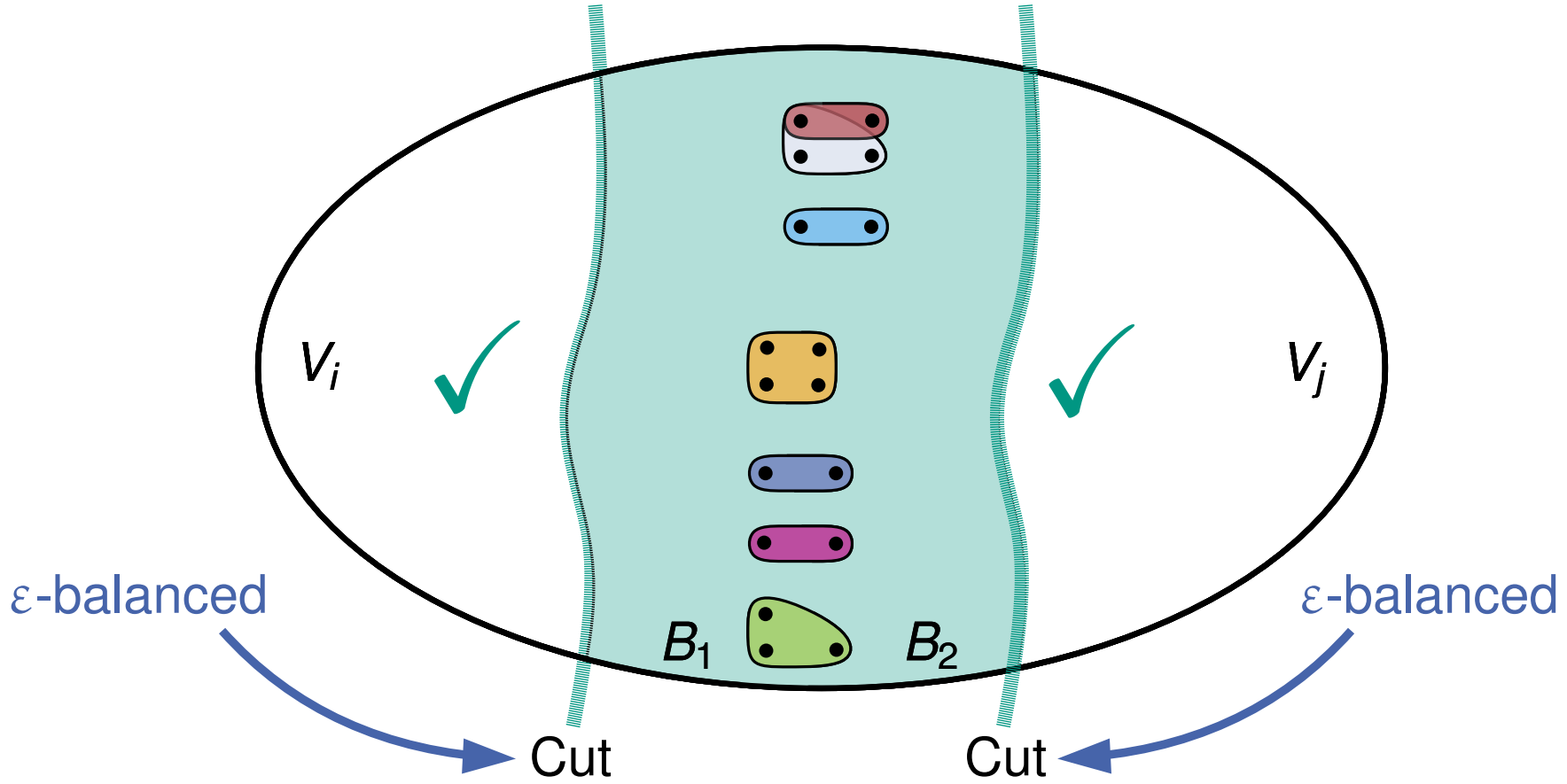
construct area $B = B_1 \cup B_2$ s.t. **every** (s,t)-cut is ε-**balanced** in H



$$c(B_1) \leq (1+\epsilon)\lceil \frac{c(V)}{k} \rceil - c(V_j) \Leftarrow \qquad \Rightarrow c(B_2) \leq (1+\epsilon)\lceil \frac{c(V)}{k} \rceil - c(V_i)$$

KaFFPa's Flow-Based Refinement for hypergraphs

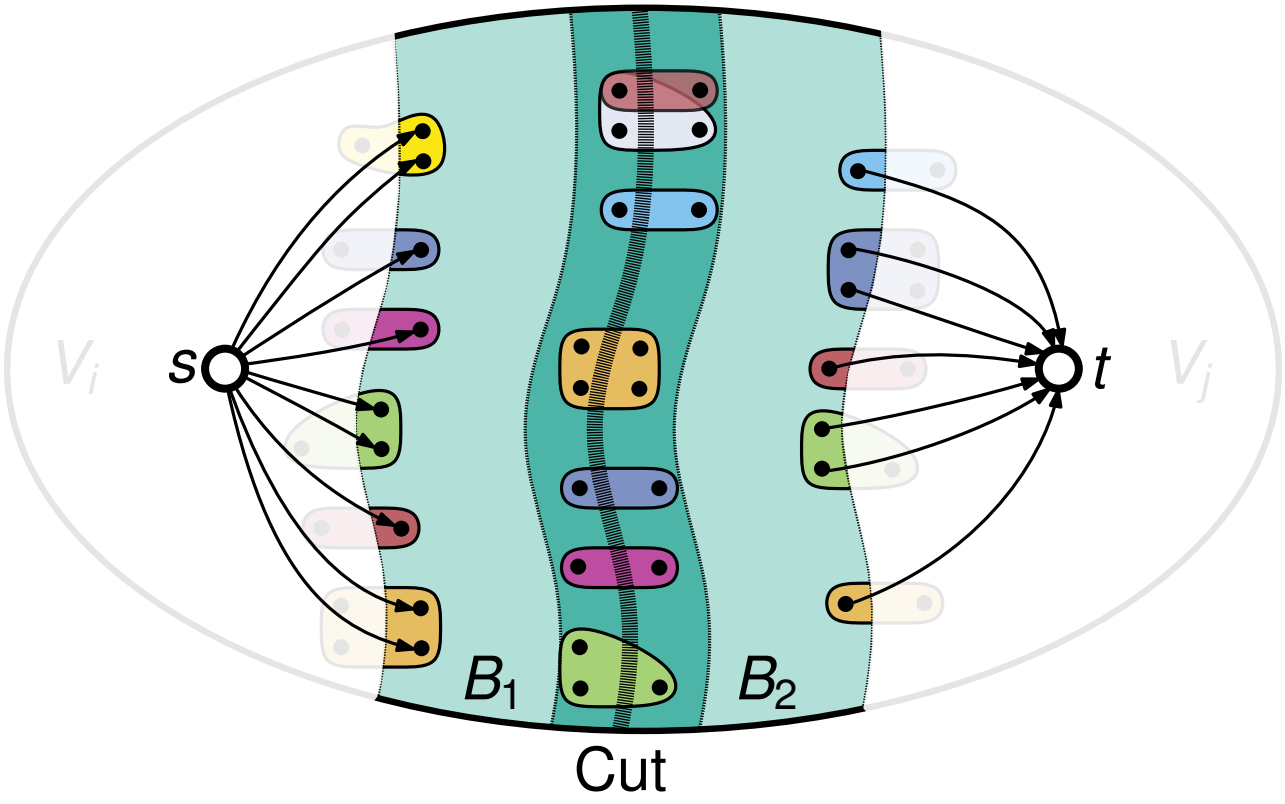
construct area $B = B_1 \cup B_2$ s.t. **every** (s,t)-cut is ε-**balanced** in H



$$c(B_1) \leq (1+\epsilon)\left\lceil \frac{c(V)}{k} \right\rceil - c(V_j) \Leftarrow \qquad \Rightarrow c(B_2) \leq (1+\epsilon)\left\lceil \frac{c(V)}{k} \right\rceil - c(V_i)$$

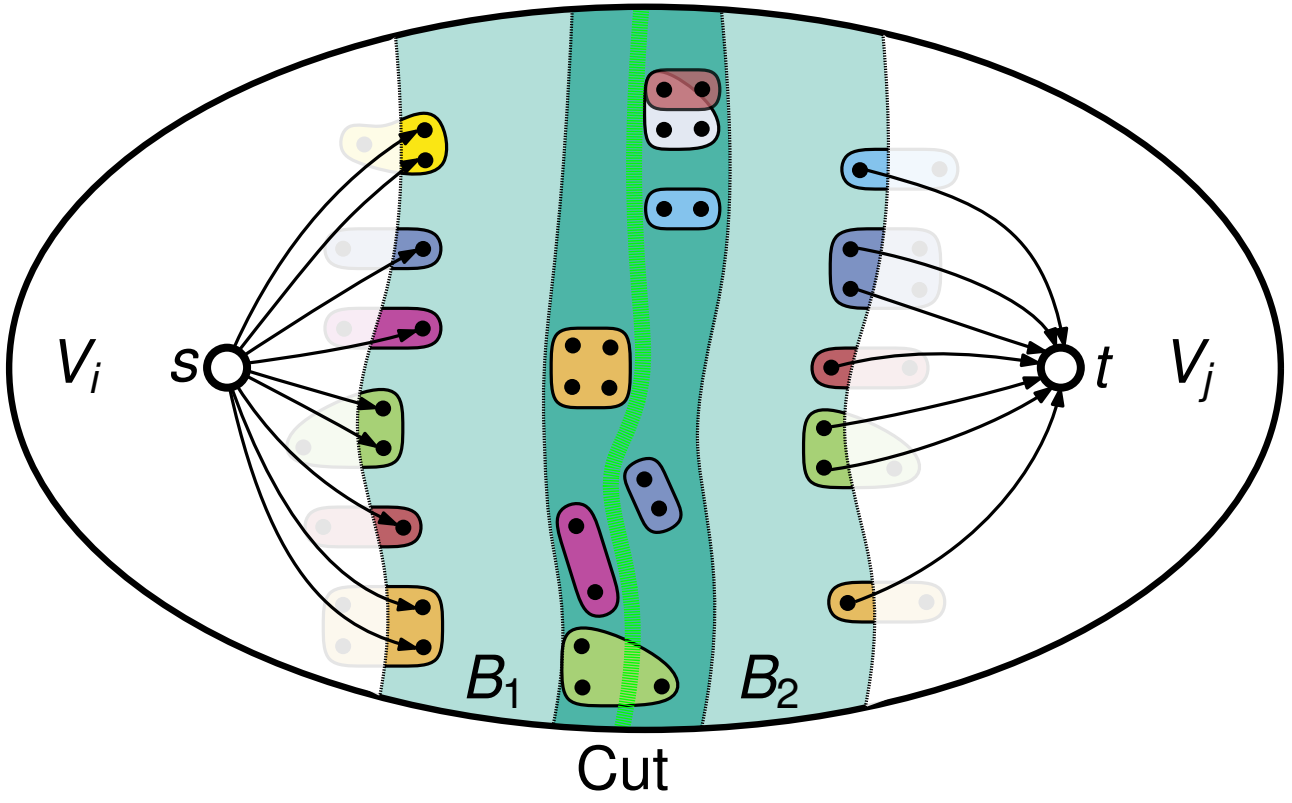
KaFFPa's Flow-Based Refinement for hypergraphs

build and solve flow problem



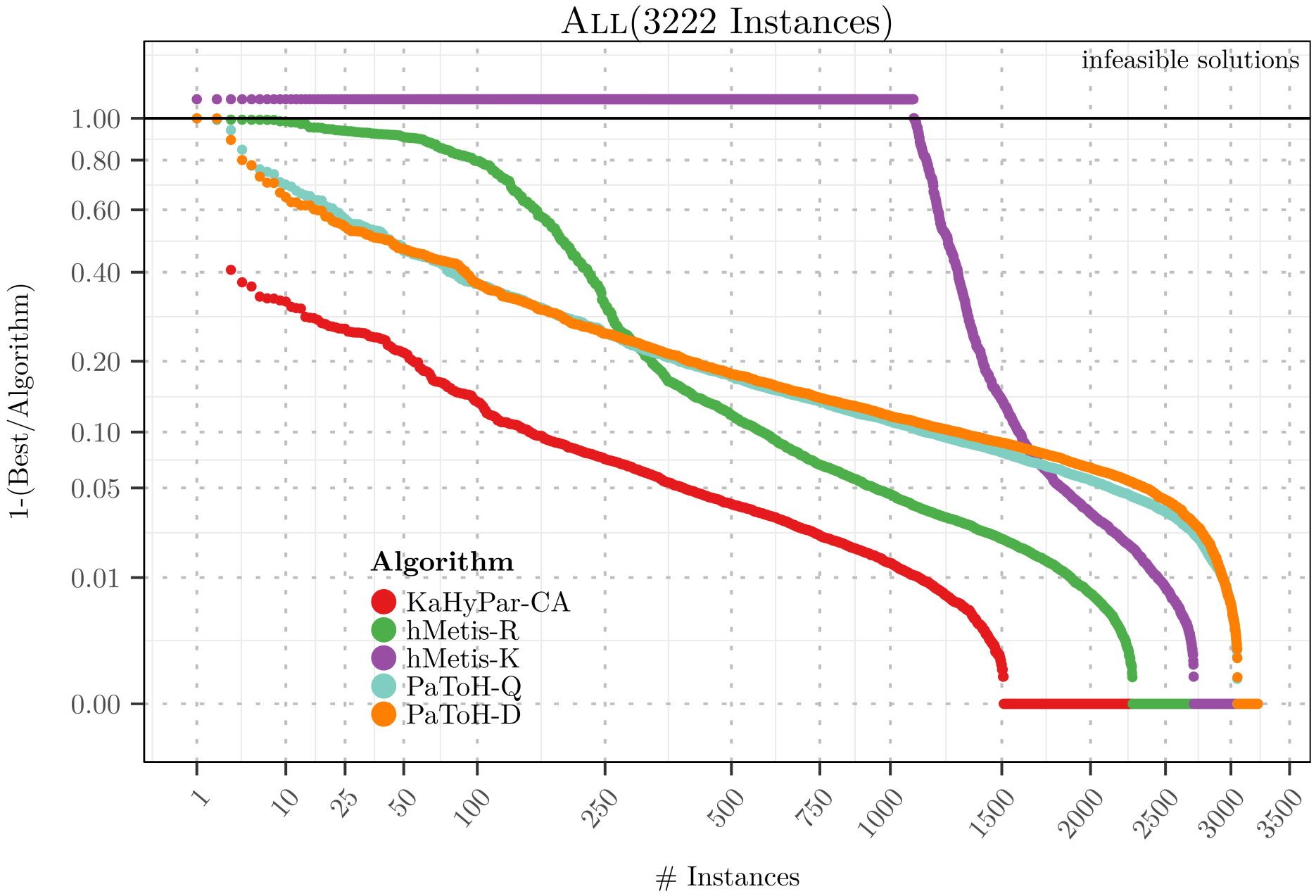
KaFFPa's Flow-Based Refinement for hypergraphs

build and solve flow problem

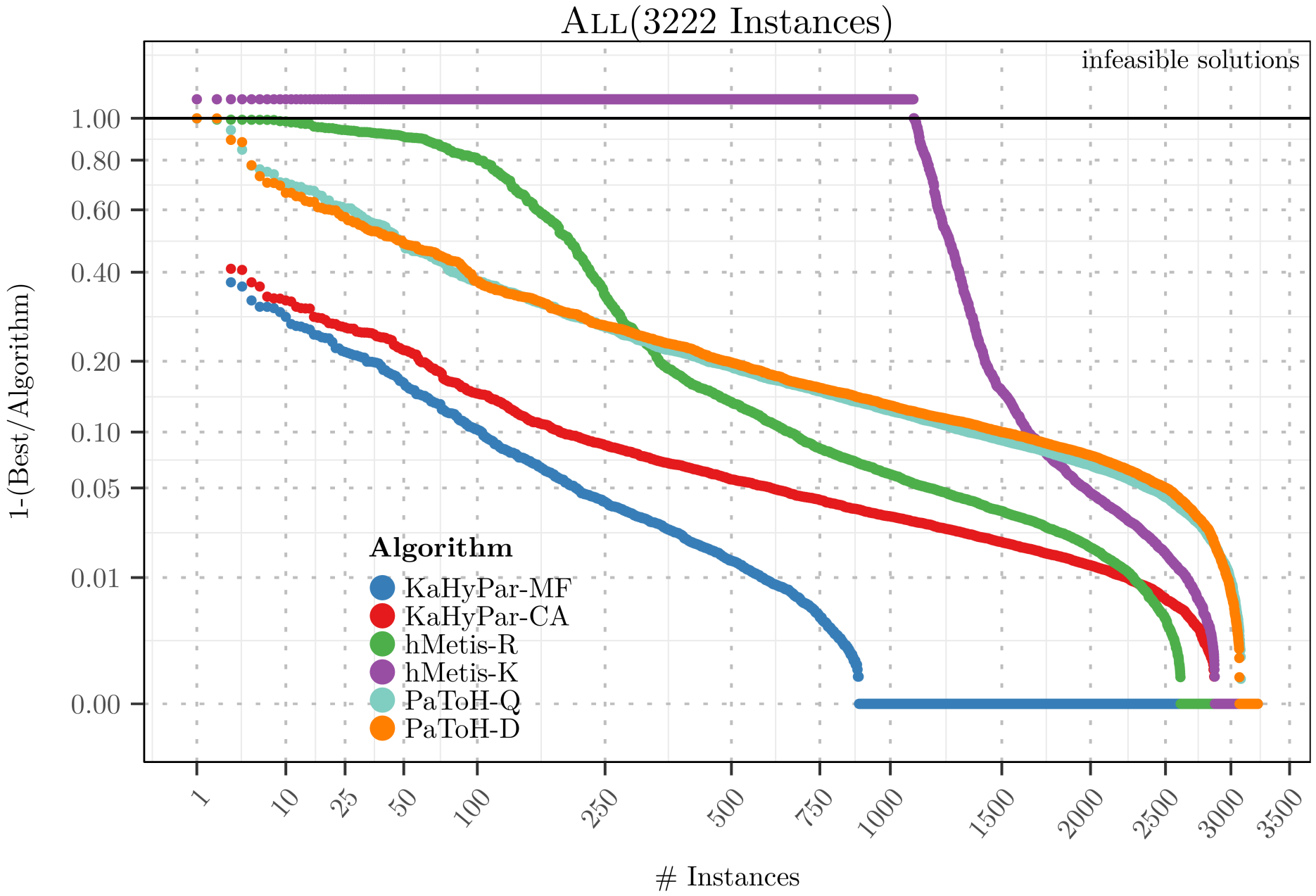


$\Rightarrow$ **optimal cut** in subhypergraph $\rightsquigarrow$ **improved** ε-balanced cut in H

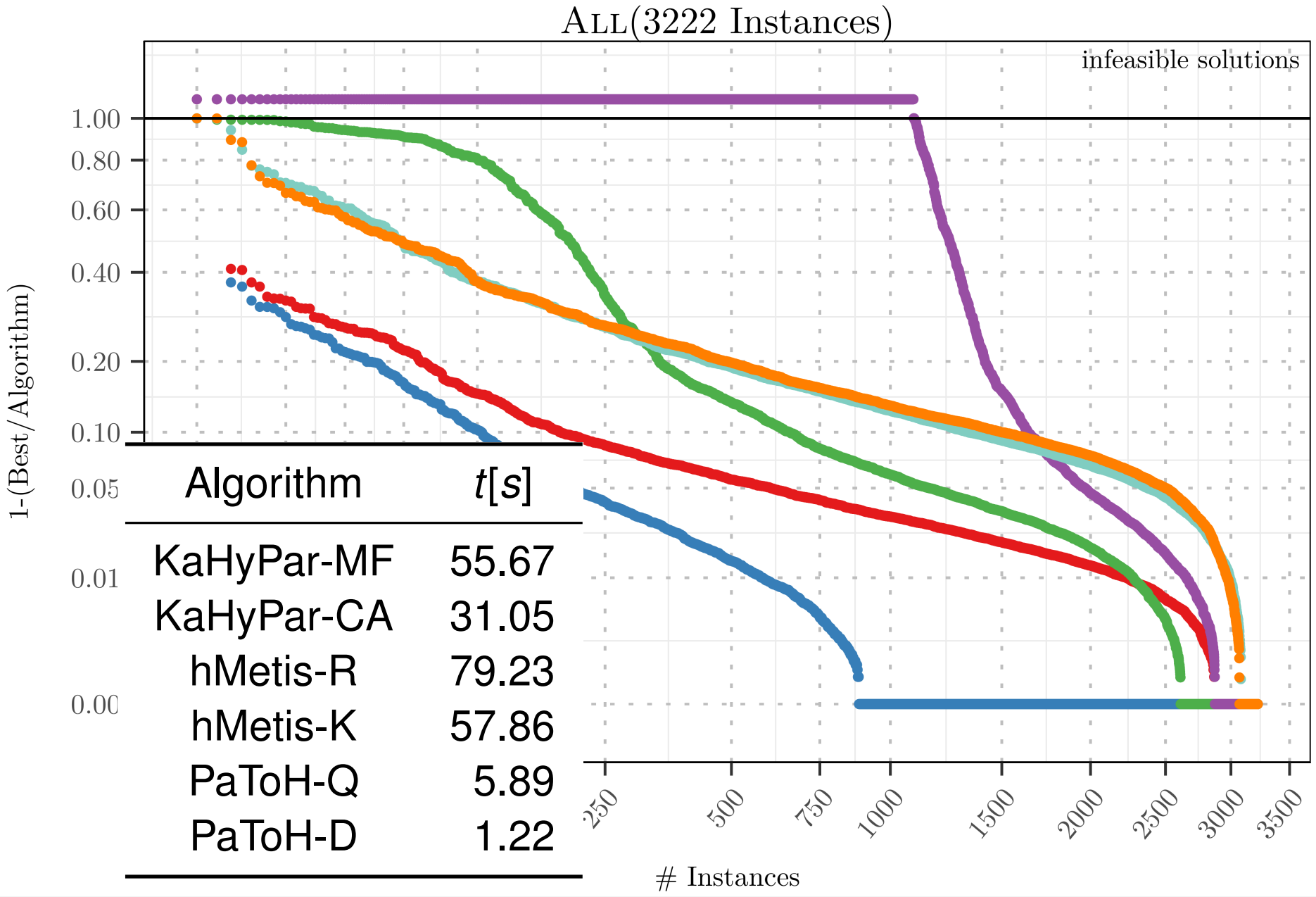
State-of-the-Art: HGP Quality



KaHyPar-MF: HGP Quality

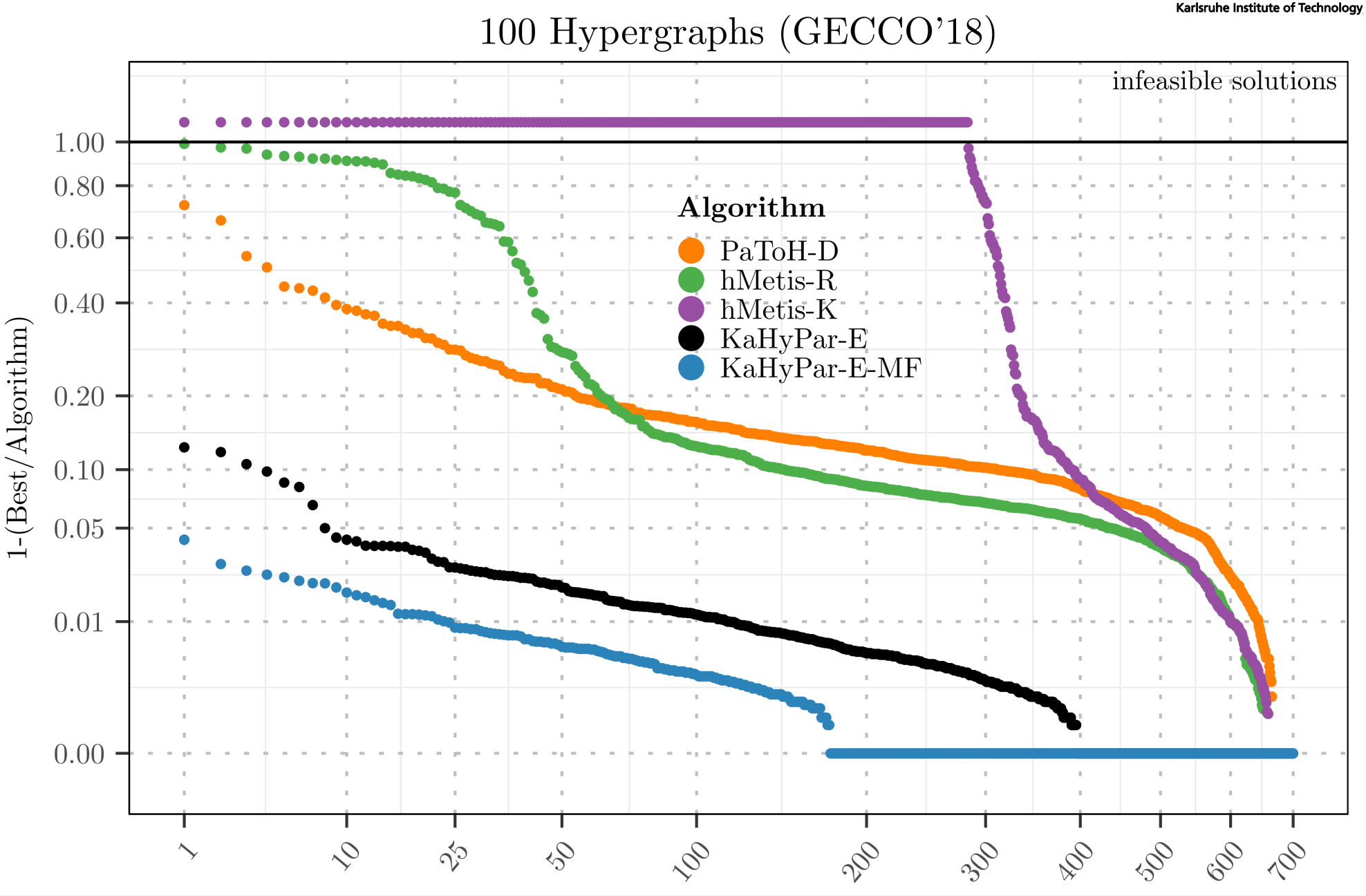


KaHyPar-MF: HGP Quality & Running Time



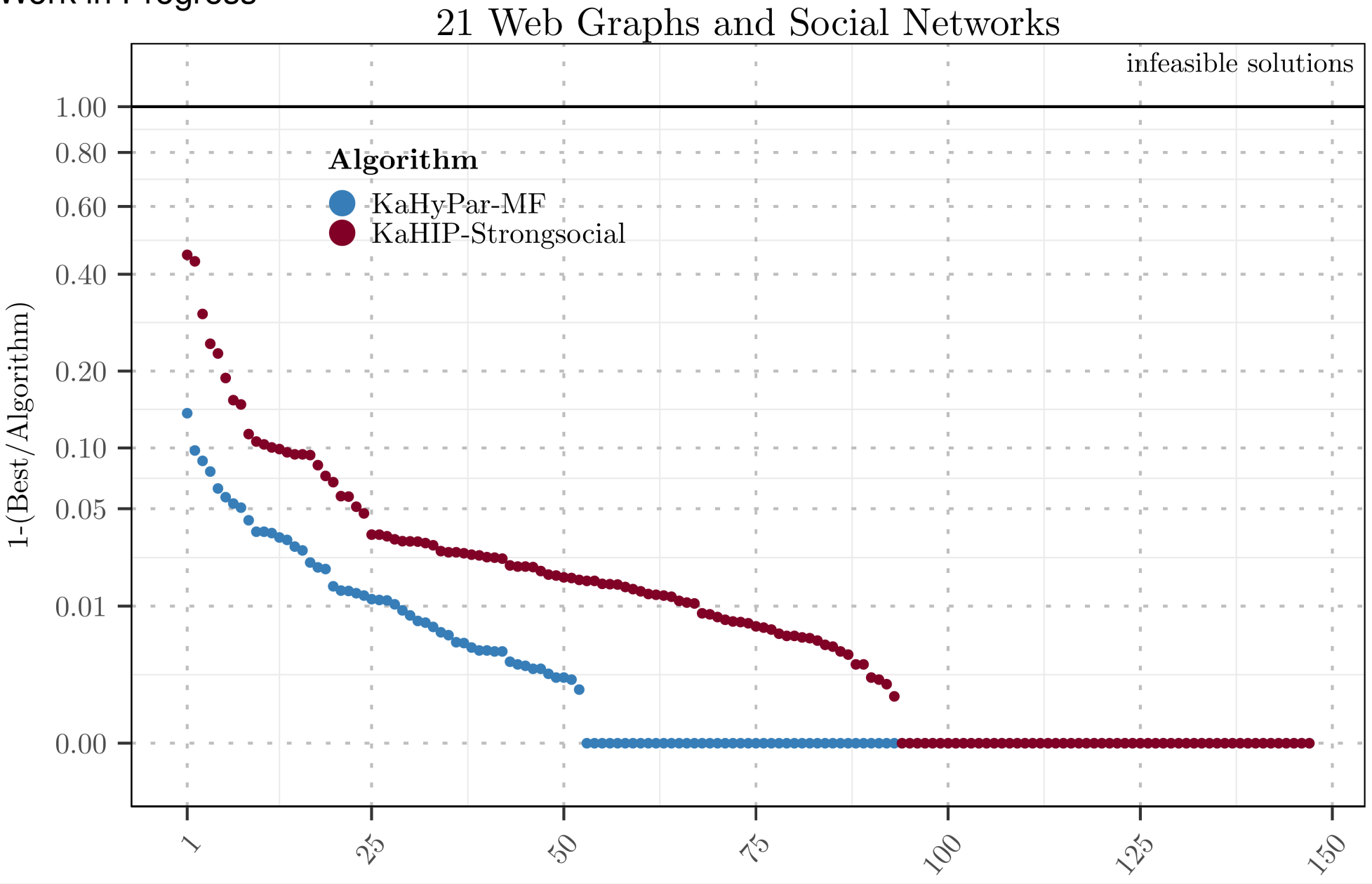
Algorithm	$t[s]$
KaHyPar-MF	55.67
KaHyPar-CA	31.05
hMetis-R	79.23
hMetis-K	57.86
PaToH-Q	5.89
PaToH-D	1.22

KaHyPar-E-MF: Flows & Memetic Algorithm



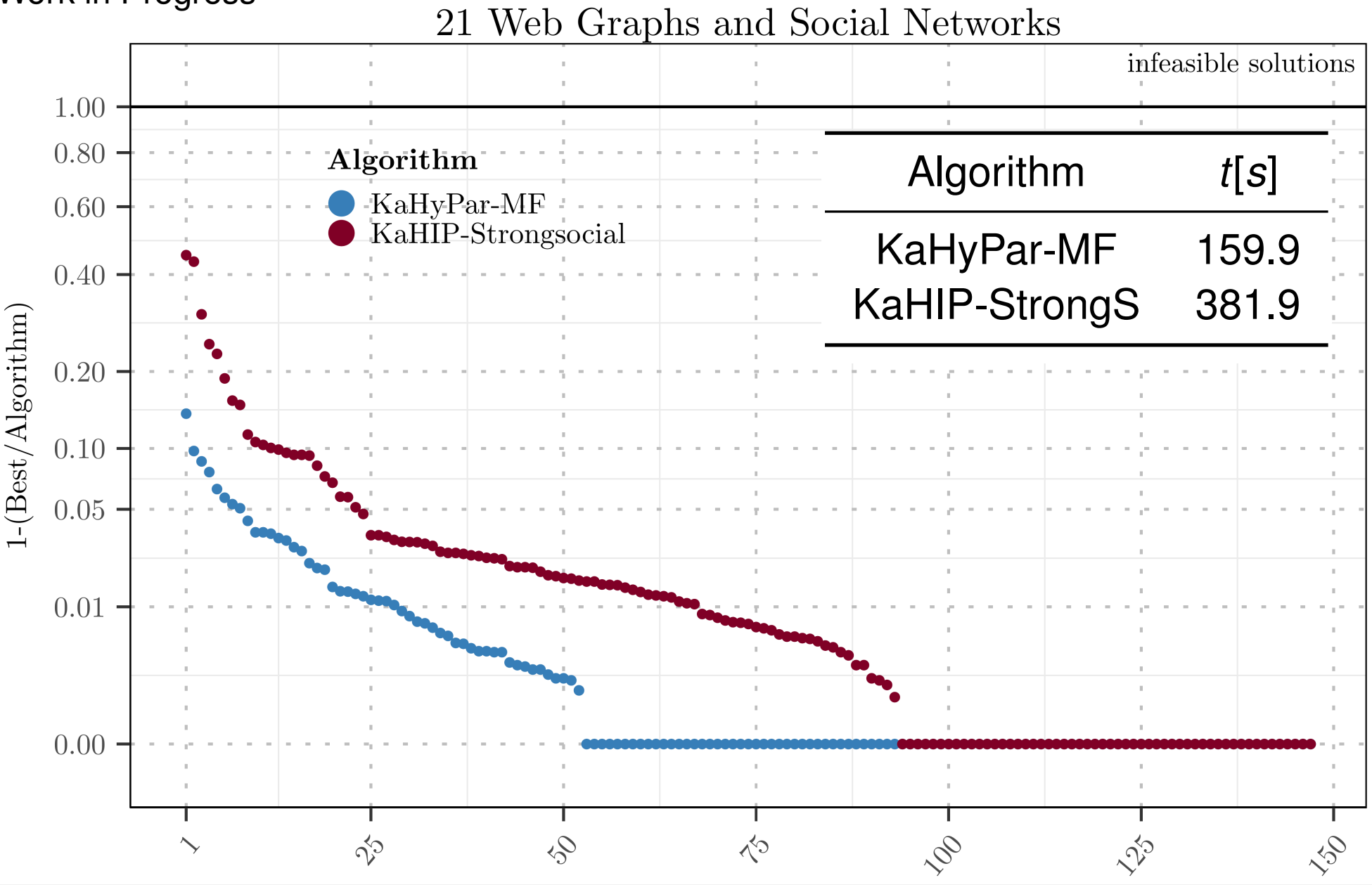
(Ab)using KaHyPar for GP

Work in Progress



(Ab)using KaHyPar for GP

Work in Progress



Algorithm	*t*[*s*]
KaHyPar-MF	159.9
KaHIP-StrongS	381.9

News on (Hyper)graph Partitioning

- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

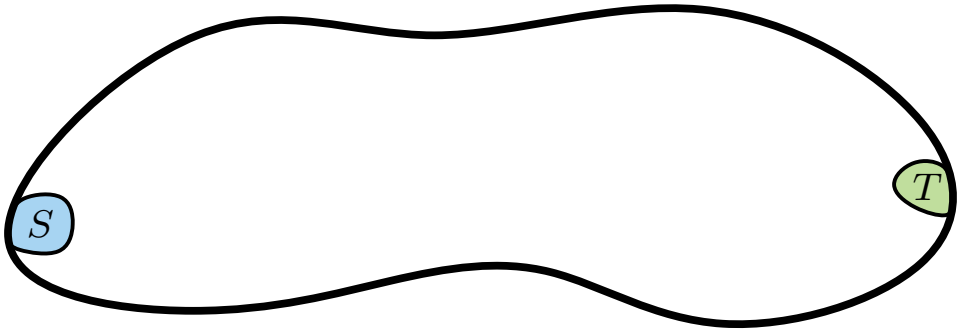
News on (Hyper)graph Partitioning

- **Highly Balanced Hypergraph Bipartitioning via Minimum *s*-*t* Cuts**
  submitted to ALENEX'19, L. Gottesbüren, M. Hamann, D. Wagner
- **Scalable Edge Partitioning**
  submitted to ALENEX'19 (TR on arXiv), S. Schlag, C. Schulz, D. Seemaier, D. Strash
- **High-Quality Shared-Memory Graph Partitioning**
  Euro-Par'18, Y. Akhremtsev, P. Sanders, C. Schulz
- **Evolutionary Multilevel Acyclic Graph Partitioning**
  GECCO'18, O. Moreira, M. Popp, C. Schulz
- **Memetic Multilevel Hypergraph Partitioning**
  GECCO'18, R. Andre, S. Schlag, C. Schulz
- **Network Flow-Based Refinement for Multilevel HGP**
  SEA'18, T. Heuer, P. Sanders, S. Schlag
- **ILP-based Local Search for Graph Partitioning**
  SEA'18, A. Henzinger, A. Noe, C. Schulz

HypergraphFlowCutter

Idea: Use repeated min cuts to compute increasingly balanced bipartitions

- used for small separators in road networks [HS18]
- similar idea for hypergraphs already in 1996 [YW96]
- use labeling technique [PM03] to simulate Lawler network [Law73]



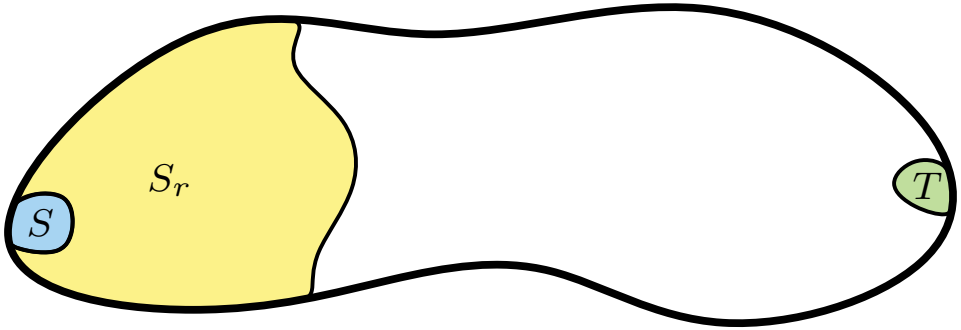
S = sources
T = targets
S_r/T_r vertices reachable from S/T wrt current max flow

[L. Gottesbüren]

HypergraphFlowCutter

Idea: Use repeated min cuts to compute increasingly balanced bipartitions

- used for small separators in road networks [HS18]
- similar idea for hypergraphs already in 1996 [YW96]
- use labeling technique [PM03] to simulate Lawler network [Law73]



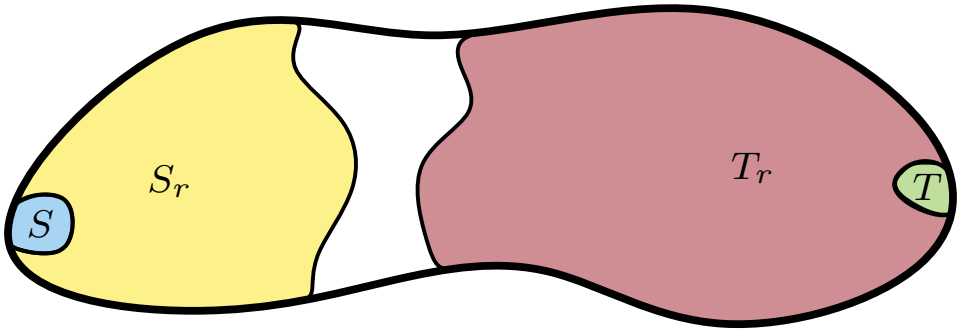
S = sources
T = targets
S_r/T_r vertices reachable from S/T wrt current max flow

[L. Gottesbüren]

HypergraphFlowCutter

Idea: Use repeated min cuts to compute increasingly balanced bipartitions

- used for small separators in road networks [HS18]
- similar idea for hypergraphs already in 1996 [YW96]
- use labeling technique [PM03] to simulate Lawler network [Law73]



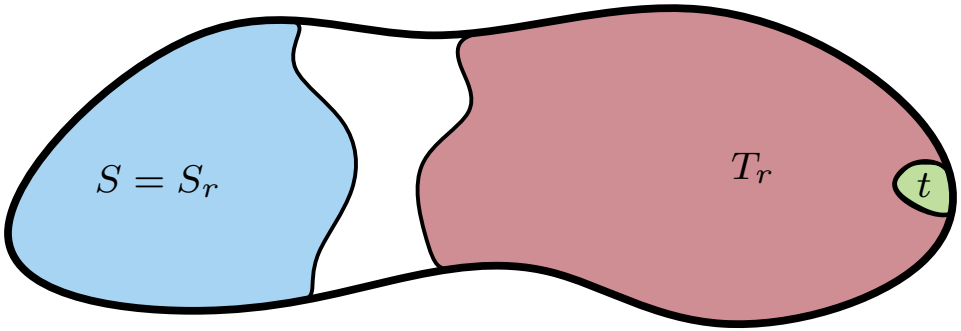
S = sources
T = targets
S_r/T_r vertices reachable from S/T wrt current max flow

[L. Gottesbüren]

HypergraphFlowCutter

Idea: Use repeated min cuts to compute increasingly balanced bipartitions

- used for small separators in road networks [HS18]
- similar idea for hypergraphs already in 1996 [YW96]
- use labeling technique [PM03] to simulate Lawler network [Law73]



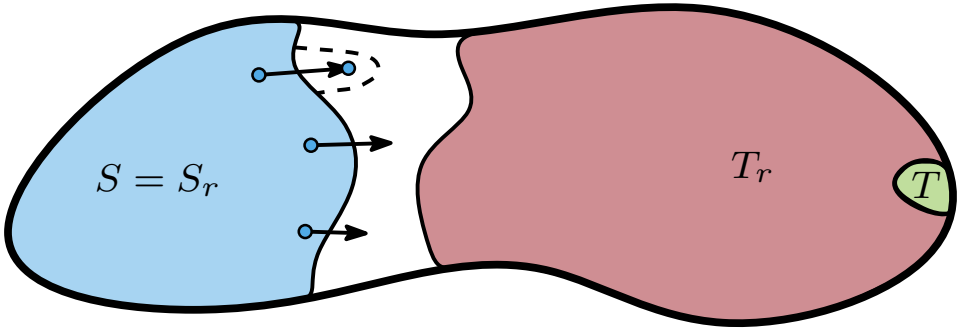
S = sources
T = targets
S_r/T_r vertices reachable from S/T wrt current max flow

[L. Gottesbüren]

HypergraphFlowCutter

Idea: Use repeated min cuts to compute increasingly balanced bipartitions

- used for small separators in road networks [HS18]
- similar idea for hypergraphs already in 1996 [YW96]
- use labeling technique [PM03] to simulate Lawler network [Law73]



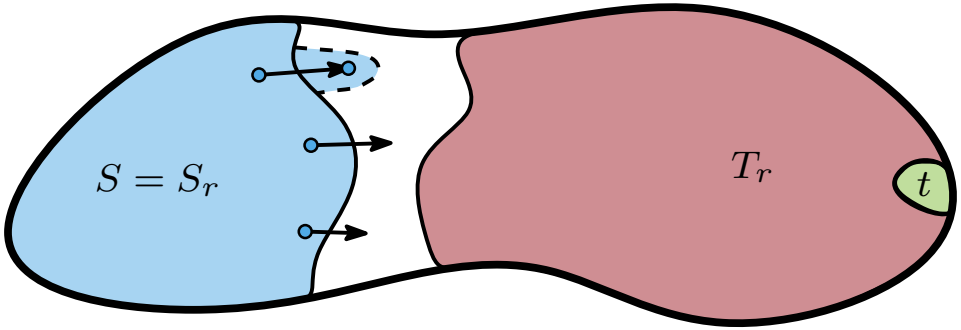
S = sources
T = targets
S_r/T_r vertices reachable from S/T wrt current max flow

[L. Gottesbüren]

HypergraphFlowCutter

Idea: Use repeated min cuts to compute increasingly balanced bipartitions

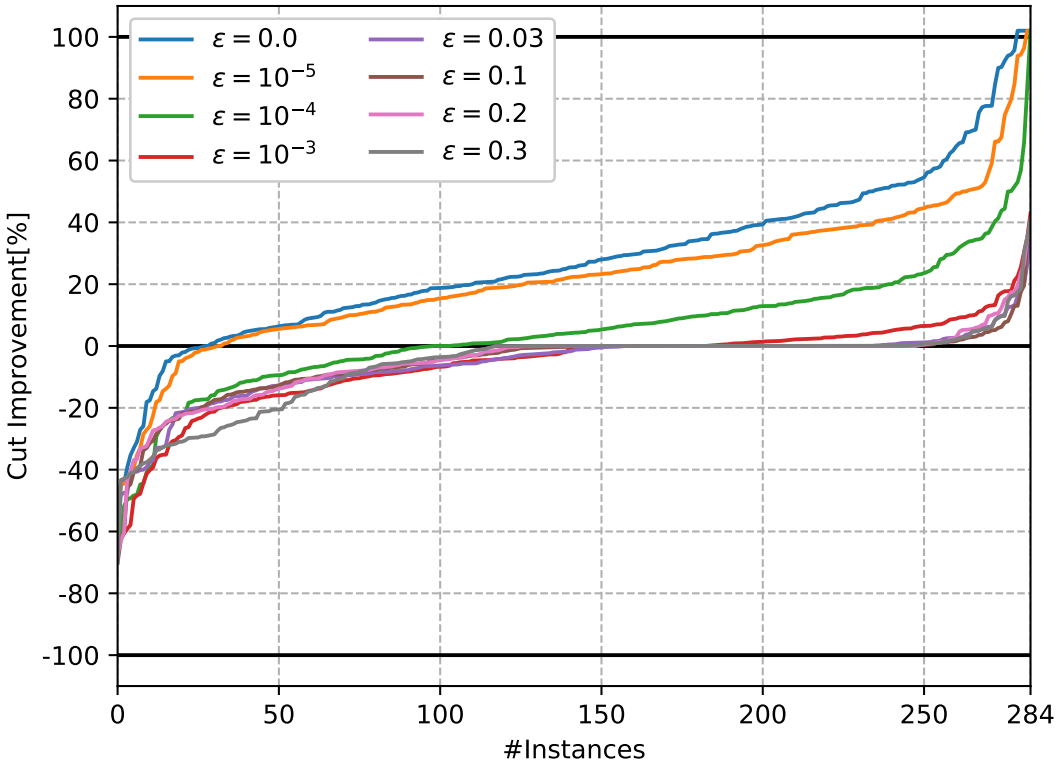
- used for small separators in road networks [HS18]
- similar idea for hypergraphs already in 1996 [YW96]
- use labeling technique [PM03] to simulate Lawler network [Law73]



S = sources
T = targets
S_r/T_r vertices reachable from S/T wrt current max flow

[L. Gottesbüren]

HyperFlowCutter - Experimental Results



ε	Running Time [s]	
	HFC-100	KaHyPar-MF
0.0	1309.6	10.4
10^{-5}	1309.5	16.0
10^{-4}	1308.3	16.3
10^{-3}	1301.6	12.5
0.03	1266.1	21.1
0.1	1211.0	24.4
0.2	1146.2	24.8
0.3	1103.7	24.4

- High running times already observed for FlowCutter on graphs
- Too slow $\Rightarrow$ use for initial partitioning. Ongoing Bachelor's thesis.

[L. Gottesbüren]

Open Source Software

(Hyper)graph Partitioning:

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