Supervised and unsupervised comparison of 30 community detection algorithms reveal that proximity to globally optimal partitions matters

Samin Aref (University of Toronto)
Joint work with
Mahdi Mostajabdaveh (Polytechnique Montréal) and
Hriday Chheda (University of Toronto)

Centre interdisciplinaire en modélisation mathématique de l'Université Laval
CIMMUL Seminar

2025-02-21

Mechanical & Industrial Engineering
UNIVERSITY OF TORONTO

Clustering a network

Nodes: club members (n)

Edges: interactions outside club (m)

Communities: densely connected groups of nodes with sparser between-group connections

Community assignments are predicted by an optimization algorithm.

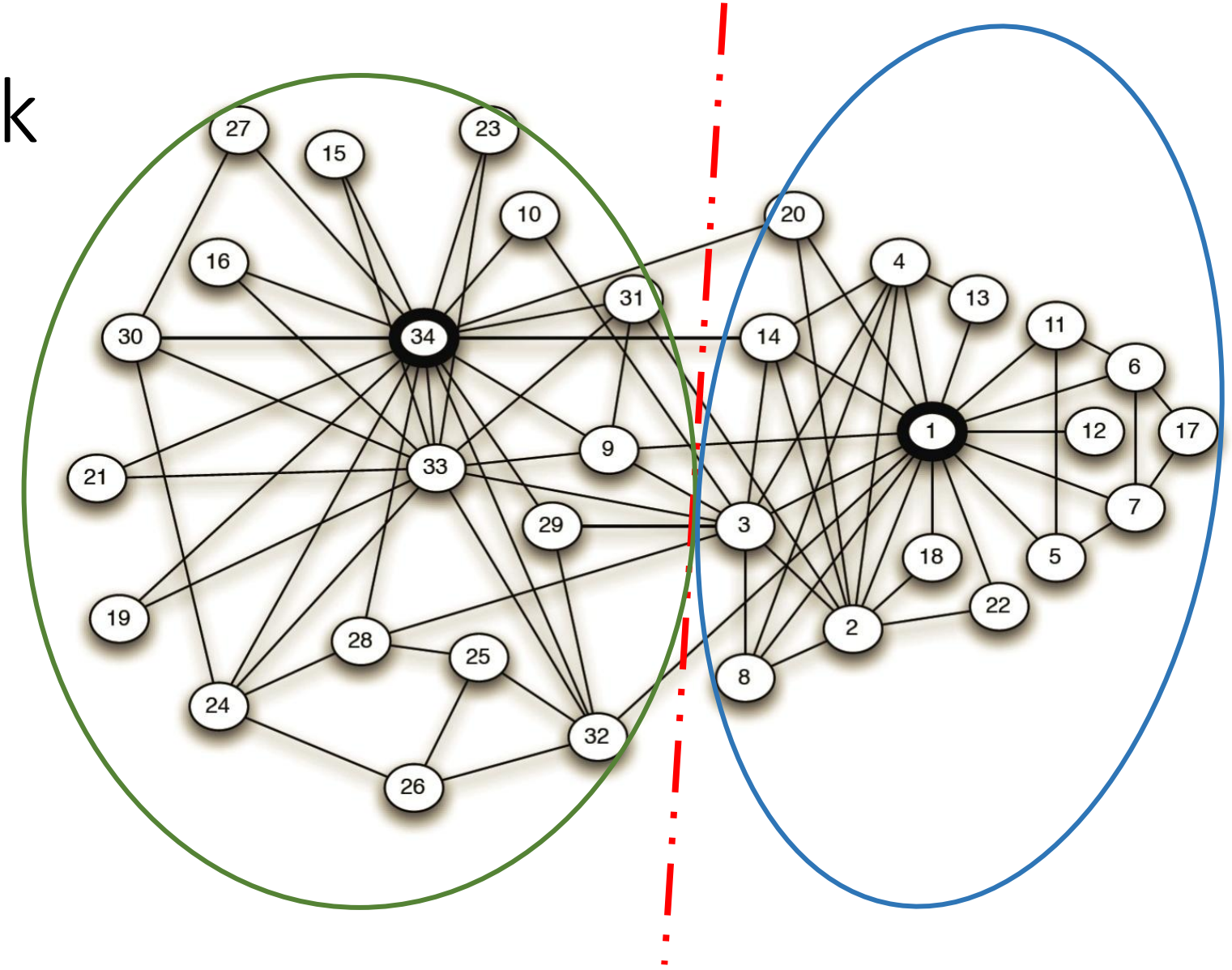


Figure from Leto Peel

Background

- There are varying degrees to the success of different algorithmic attempts at optimizing any function.

- Over the past 15-20 years, the term **modularity maximization (MM)** has been used for referring to **attempts at maximizing modularity** (whose success rates have remained unknown).

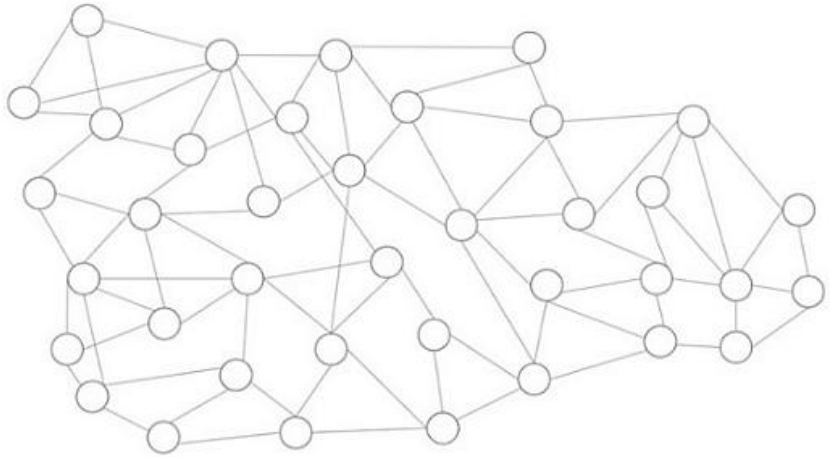
- If those MM attempts fail more than they succeed, we actually do not know much about maximum modularity partitions (for better or worse).

UNIVERSITY OF TORONTO

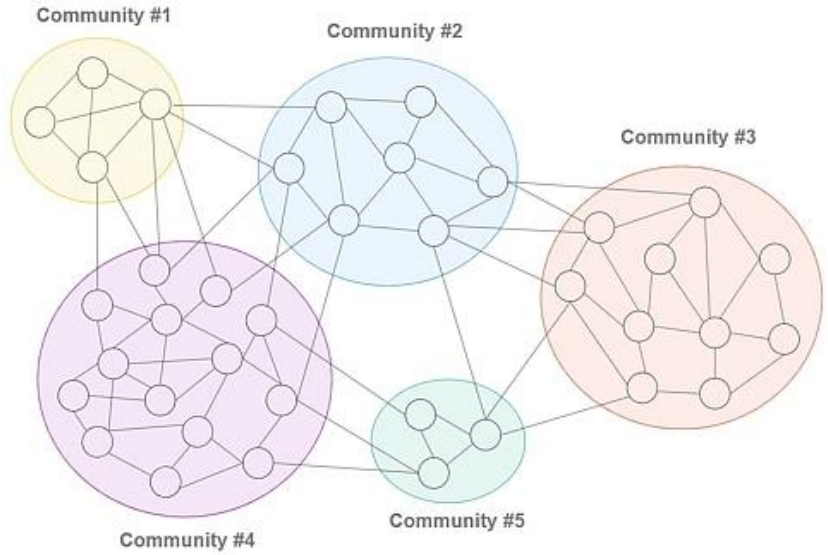
Problem definition

- Input: graph $G(V,E)$ with n nodes and m edges
- Method: a clustering (descriptive community detection) algorithm
- Output: a partition of the nodes into communities (node colours)
- Goal: grouping tightly connected nodes into communities (clusters) which can be gently connected to the rest of the network (Schaub et al. 2017).

Input



Output



Figures from timbr.ai

Schaub et al. (2017) The many facets of community detection in complex networks

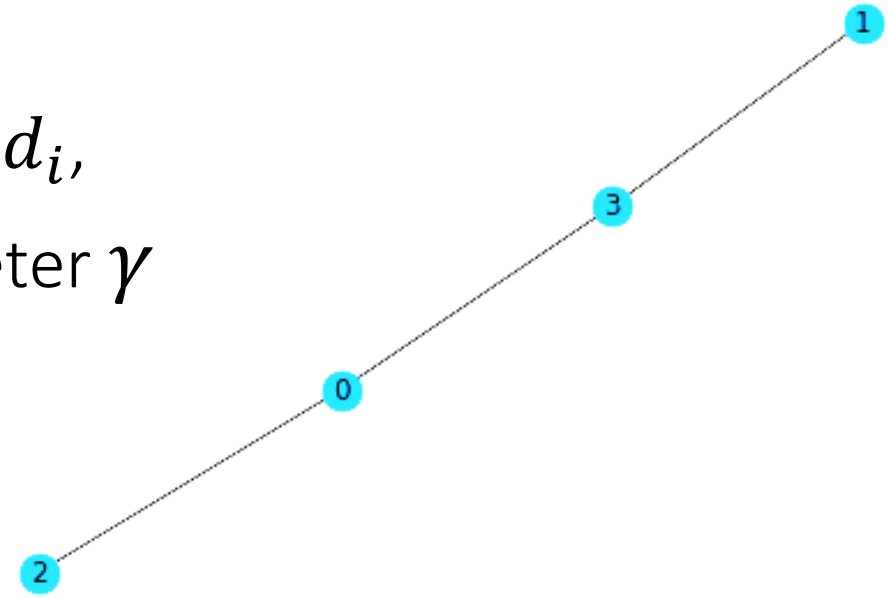
Modularity

Given a network (graph) *G(V,E),* find a partition of *V* into X={V_1, V_2, …, V_c} such that the modularity *Q(G,X)* is maximized.

Modularity entry b_{ij} : a function of degrees d_i,

connections a_{ij}, and the resolution parameter γ

$$b_{ij} = a_{ij} - \gamma \frac{d_i d_j}{2m}$$



UNIVERSITY OF TORONTO

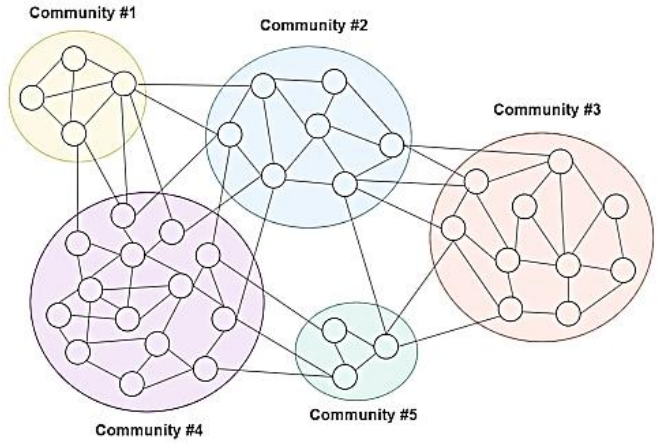
Detecting communities via Modularity Maximization (MM)

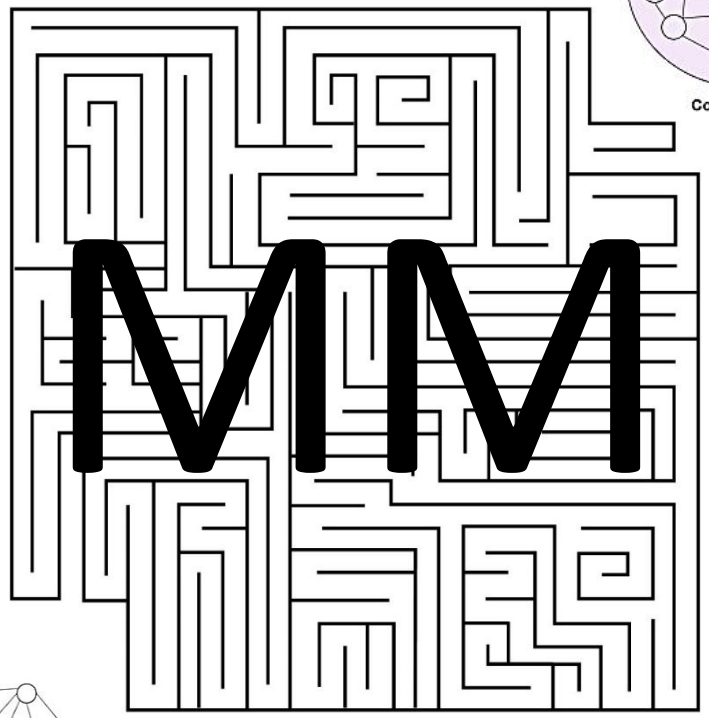
Given graph *G(V,E)* with modularity entries b_{ij}, find a partition of the node set *V* into $X=\{V_1, V_2, \ldots, V_k\}$ to maximizes the objective function *Q(G,X)*.

$$Q_{(G,X)} = \frac{1}{2m} \sum_{(i,j)\in V^2} b_{ij}$$

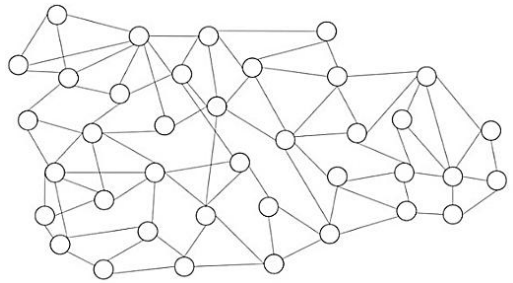
i and j are together in partition X

Output: communities





Input: a simple graph

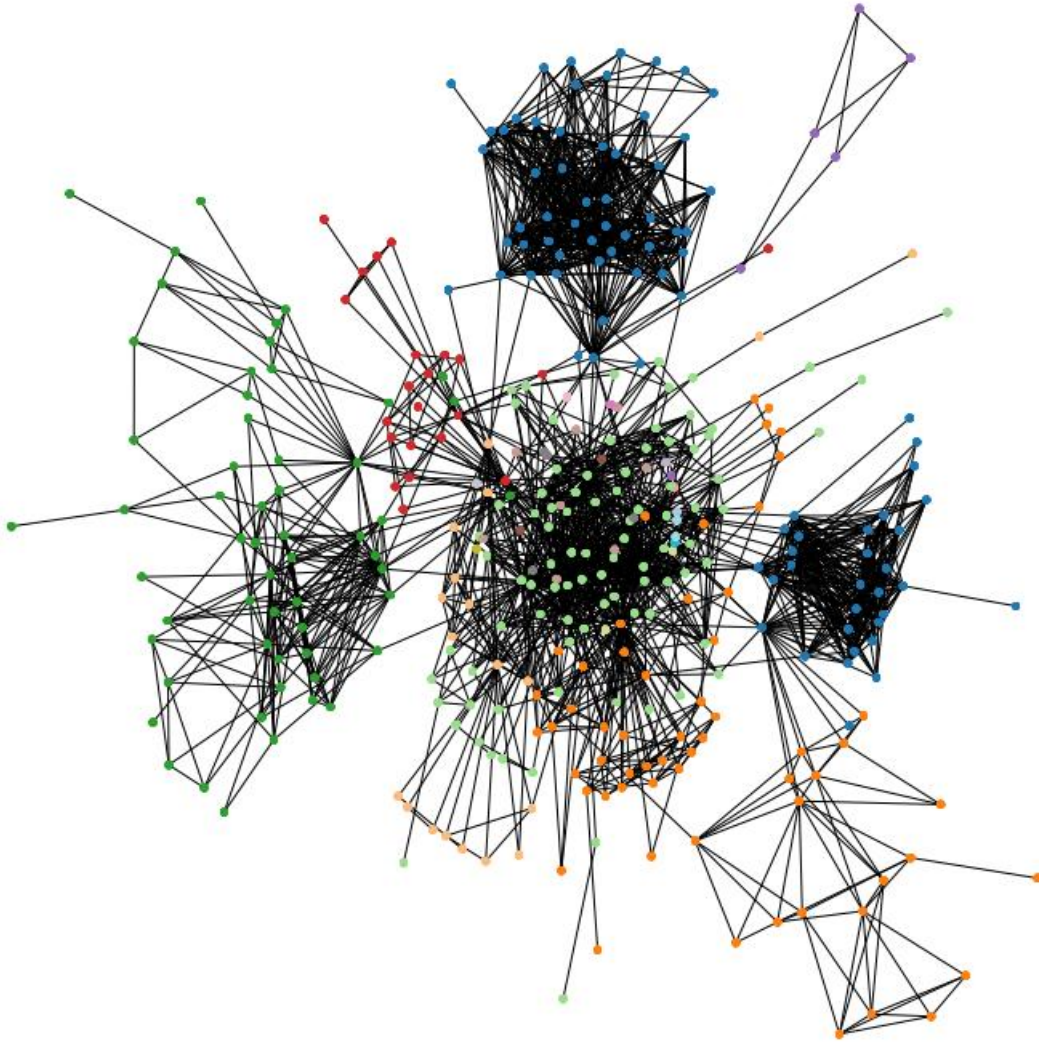


Figures from timbr.ai

Communities depend on the algorithm even if the objective function is the same.

Dataset:
facebook_friends
m=1988

Q: modularity for a partition
Q*: maximum modularity of the graph
k: number of communities
AMI: adjusted mutual information
(similarity to an optimal partition)

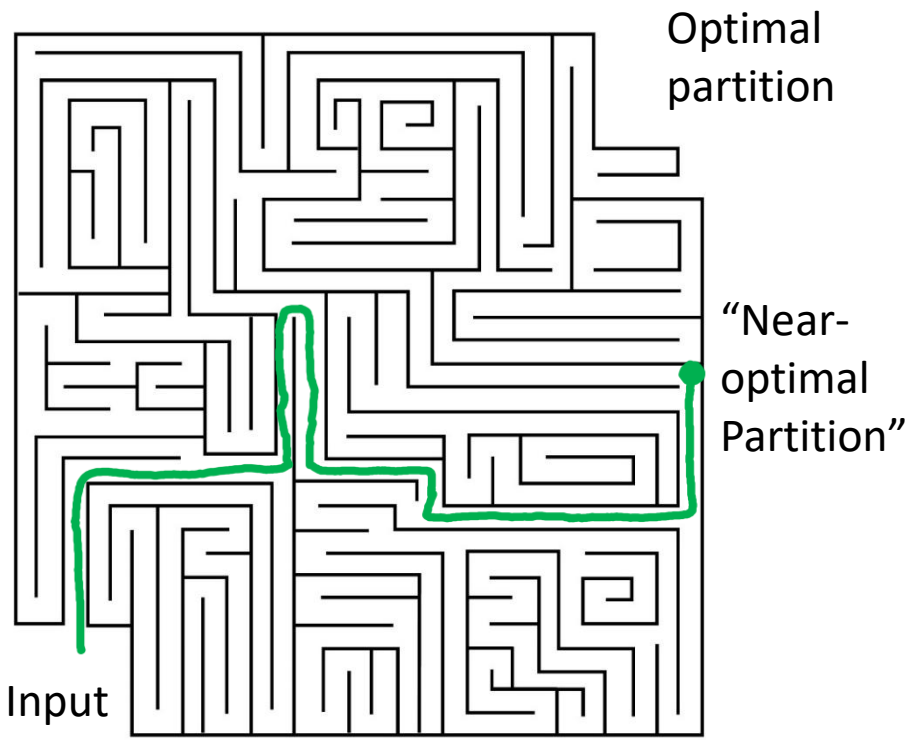


(a) Bayan, $Q^* = 0.7157714$, $k = 28$, AMI $= 1$

UNIVERSITY OF TORONTO

Approach 1: Modularity Maximization Heuristics

1. Edge Motif (EdMot) (Li et al. 2019)
2. Leiden (Traag et al. 2019)
3. Paris (Bonald et al. 2018)
4. Belief (Zhang & Moore 2014)
5. Combo (Sobolevsky et al. 2014)
6. Leicht-Newman (LN) (Leicht & Newman 2008)
7. Louvain (Blondel et al. 2008)
8. Greedy (CNM) (Clauset et al. 2004)
9. Graph Neural Network (GNN) (Sobolevsky et al. 2022)



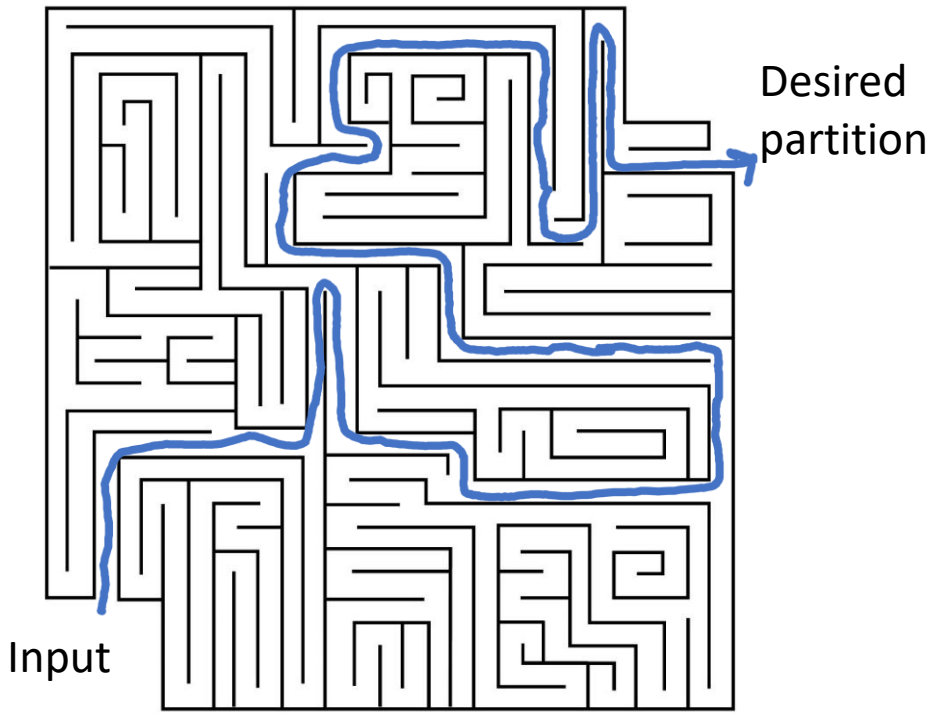
Approach 2: Exact/Approximate Modularity Maximization

- Integer Programming - IP (Brandes et al. 2007)
- IP and LP rounding (Agarwal & Kempe 2008)
- Column generation (Aloise et al. 2010)
- Sparse IP and LP rounding (Dinh & Thai 2015)
- Approximation (Kawase et al. 2021)
- Bayan (Aref et al. 2022) for graphs with m<3000

$$\max_{x_{ij}} Q = \frac{1}{2m}\left(\sum_{(i,j)\in V^2, i<j} 2b_{ij}(1-x_{ij}) + \sum_{i\in V} b_{ii}\right)$$

$$\text{s.t.} \quad x_{ik} + x_{jk} \geq x_{ij} \quad \forall (i,j) \in V^2, i<j, k \in K(i,j)$$

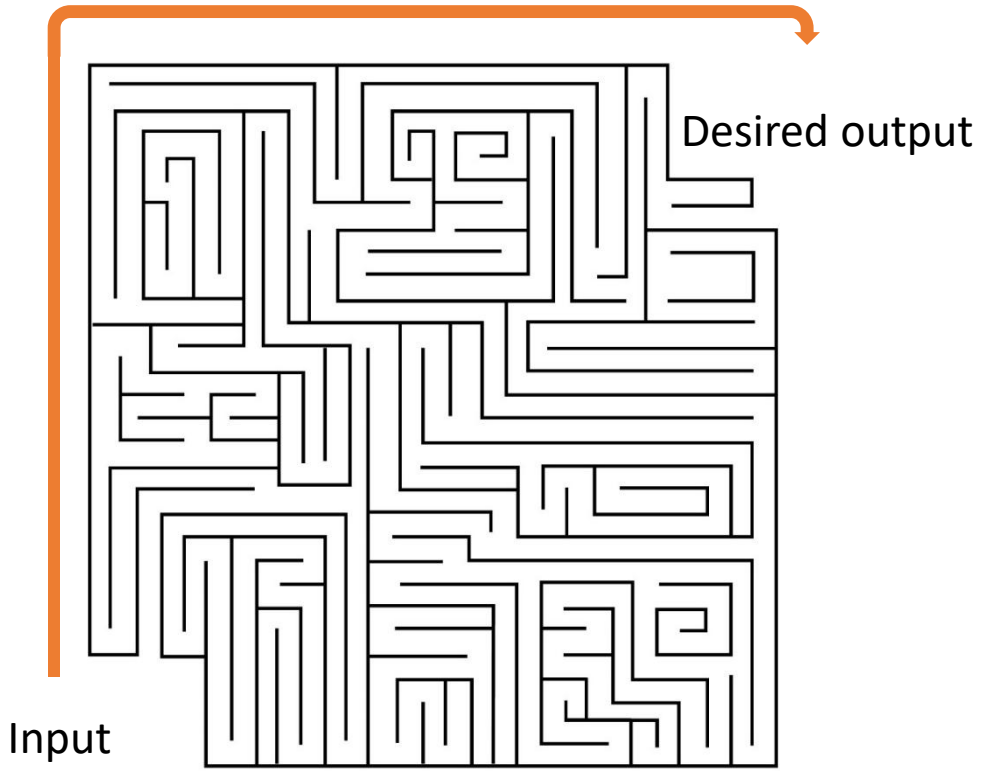
$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in V^2, i<j$$



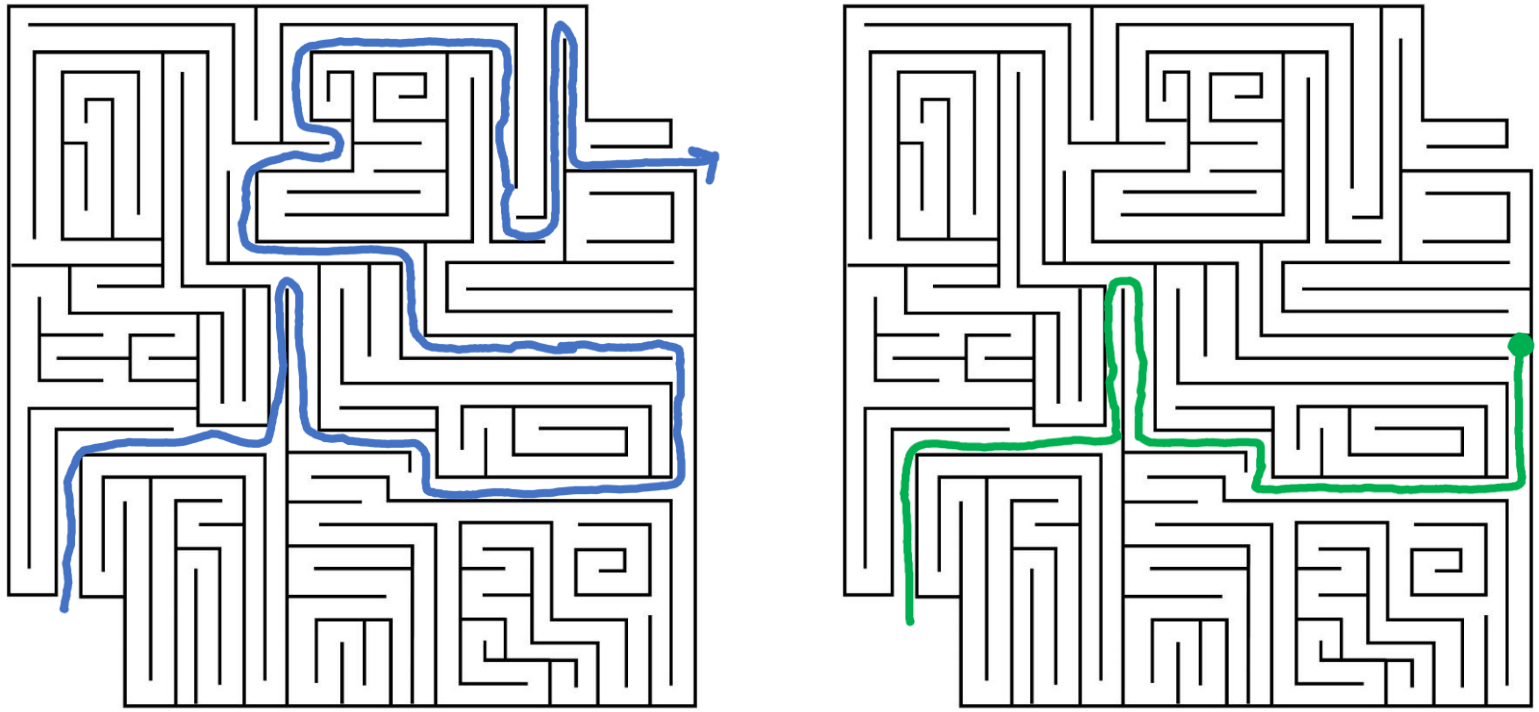
UNIVERSITY OF TORONTO

Approach 3: Other Methods

1. Kernighan-Lin bisection (Kernighan and Lin 1970)
2. RB Potts model with Erdős–Rényi as null (Reichardt & Bornholdt 2006)
3. Chinese whispers (Biemann et al. 2006)
4. Walktrap (Pons & Latapy 2006)
5. k-cut (Ruan & Zhang 2007)
6. Asynchronous label propagation (Raghavan et al. 2007)
7. Infomap (Rosvall & Bergstrom 2008)
8. Genetic Algorithm (Pizzuti 2008)
9. Semi-synchronous Label propagation (Cordasco & Gargano 2010)
10. Constant Potts Model (CPM) (Traag et al. 2011)
11. Significant scales (Traag et al. 2013)
12. Stochastic Block Model (SBM) (Peixoto 2014a)
13. SBM with Monte Carlo Markov Chain (MCMC) (Peixoto 2014b)
14. WCC (Prat-Pérez et al. 2014)
15. Surprise (Traag et al. 2015)
16. Diffusion Entropy Reducer (DER) (Kozdoba and Mannor 2015)
17. GemSec (Rozemberczki et al. 2019)
18. Bayesian Planted Partition (BPP) (Zhang and Peixoto, 2020)
19. Markov Stability (PyGenStability) (Arnaudon et al., 2023)

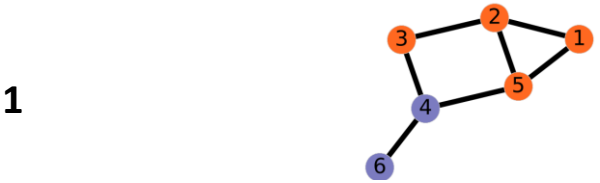
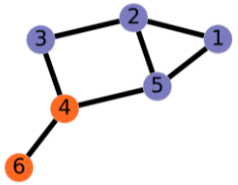
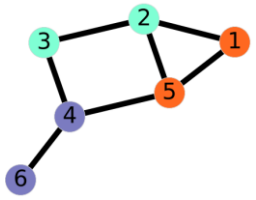
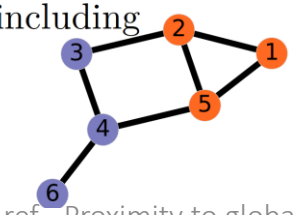


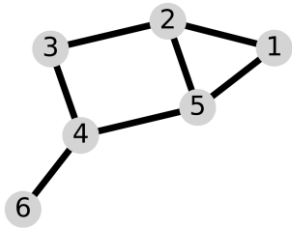
Assessing the optimality of partition among ten modularity-based algorithms



UNIVERSITY OF TORONTO

Evaluating ten modularity-based algorithms

Ten algorithms (and their variations)	Partition (obtained by an MM algorithm)	Modularity	Optimality percentage (GOP)	Maximum similarity to any optimal partition - AMI (or RMI or ECS)
Method 1		Q_1=0.122	Q_1/Q^*=60%	36%
Method 2		Q_2=0.122	Q_2/Q^*=60%	36%
...	...	...	...	...
Method 19		Q_{19}=0.092	Q_{19}/Q^*=45%	30%
Exact method (Integer Programming)	All optimal partitions including 	Q*=0.204	100%	100%
			Y-axis	X-axis

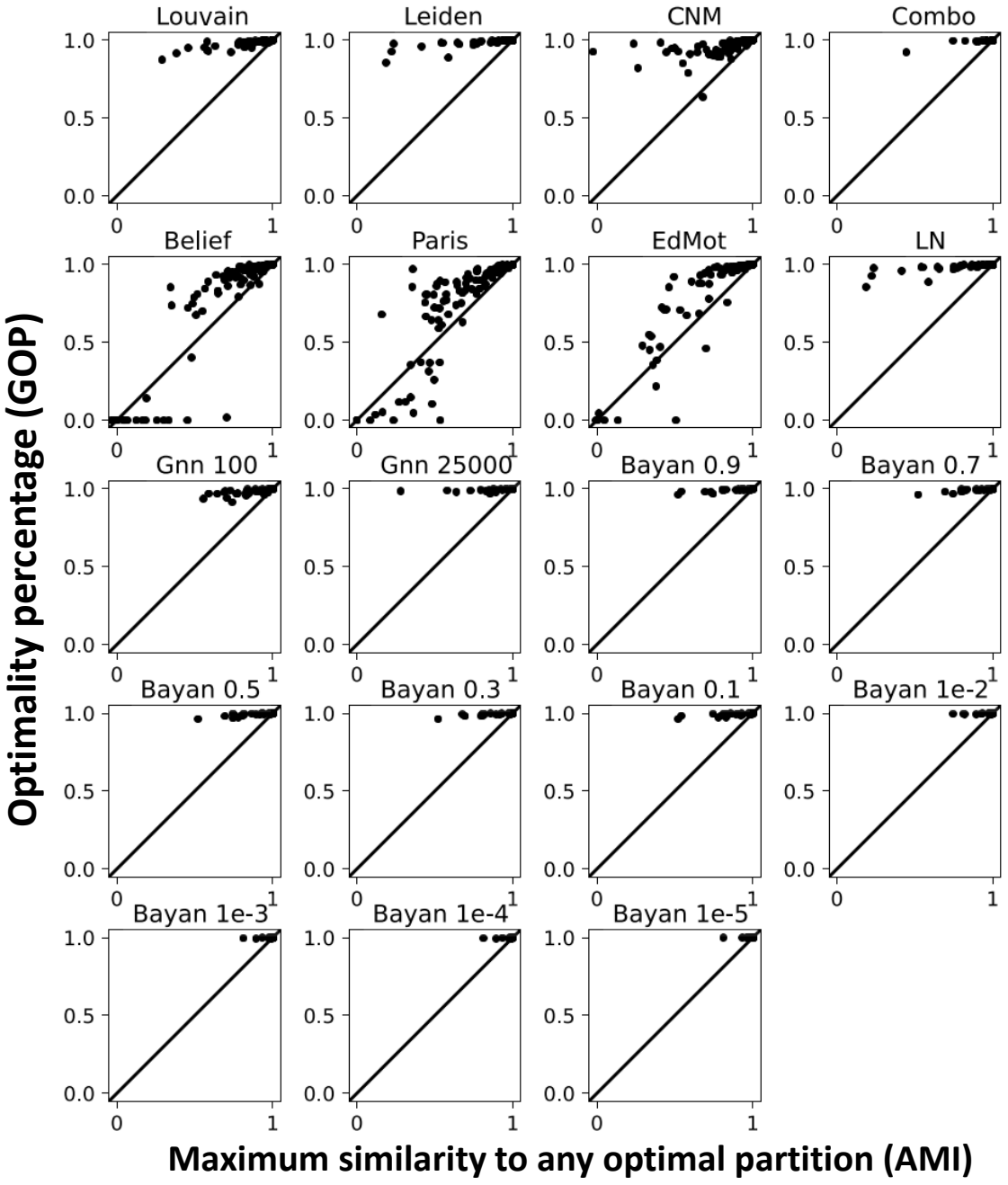


Test cases are 104 graphs with modular structure:

- 54 real networks
- 20 LFR random graphs
- 30 ABCD random graphs

Each datapoint represents the performance of one algorithm on one test case.

1. Y-values: Most partitions obtained from heuristics are sub-optimal
2. X-values: Many partitions are dissimilar to any optimal partition
3. 45°-line: near-optimal partitions are not similar to any optimal partitions



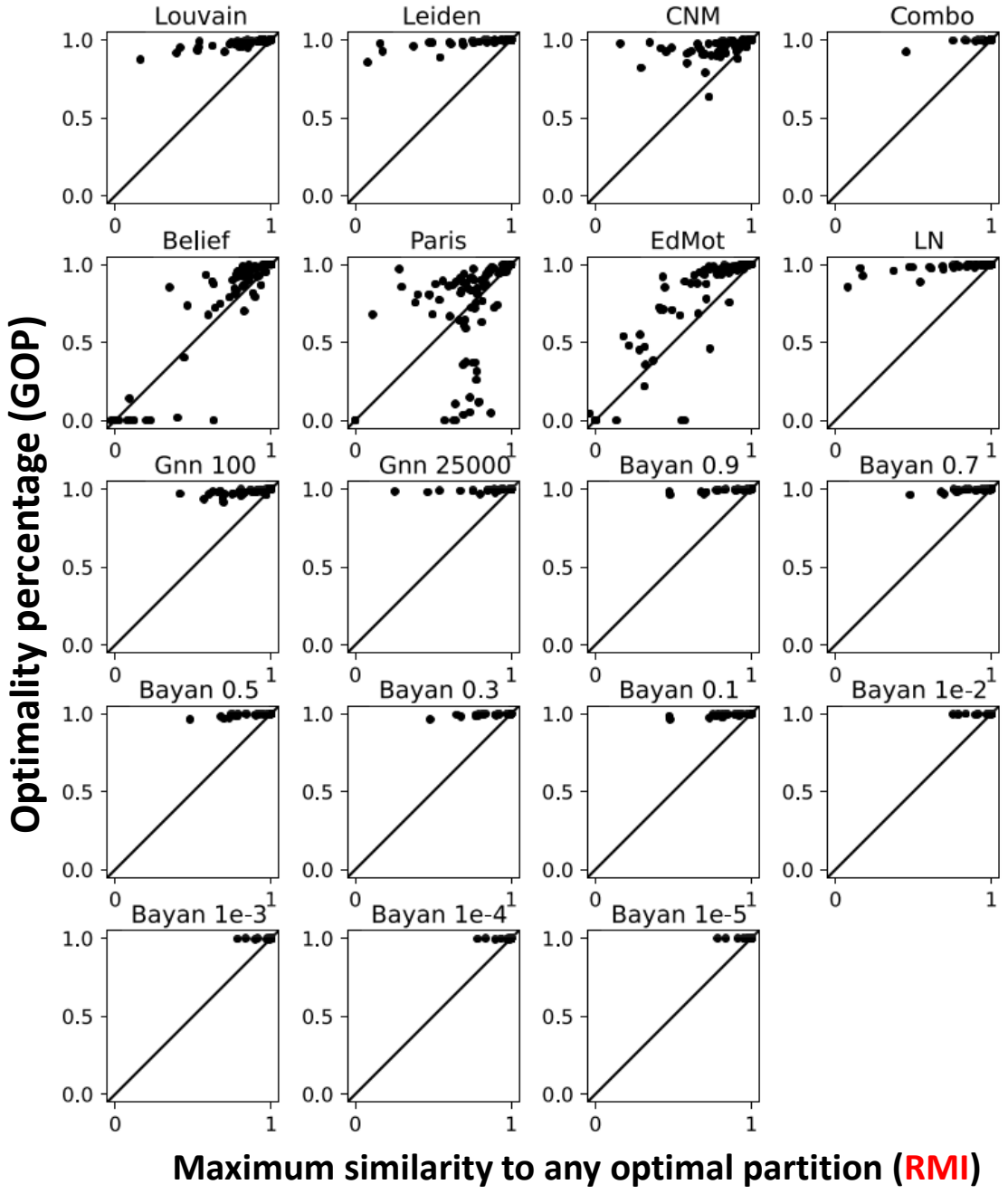
UNIVERSITY OF TORONTO

Test cases are 104 graphs with modular structure:

- 54 real networks
- 20 LFR random graphs
- 30 ABCD random graphs

Each datapoint represents the performance of one algorithm on one test case.

1. Y-values: Most partitions obtained from heuristics are sub-optimal
2. X-values: Many partitions are dissimilar to any optimal partition
3. 45°-line: near-optimal partitions are not similar to any optimal partitions



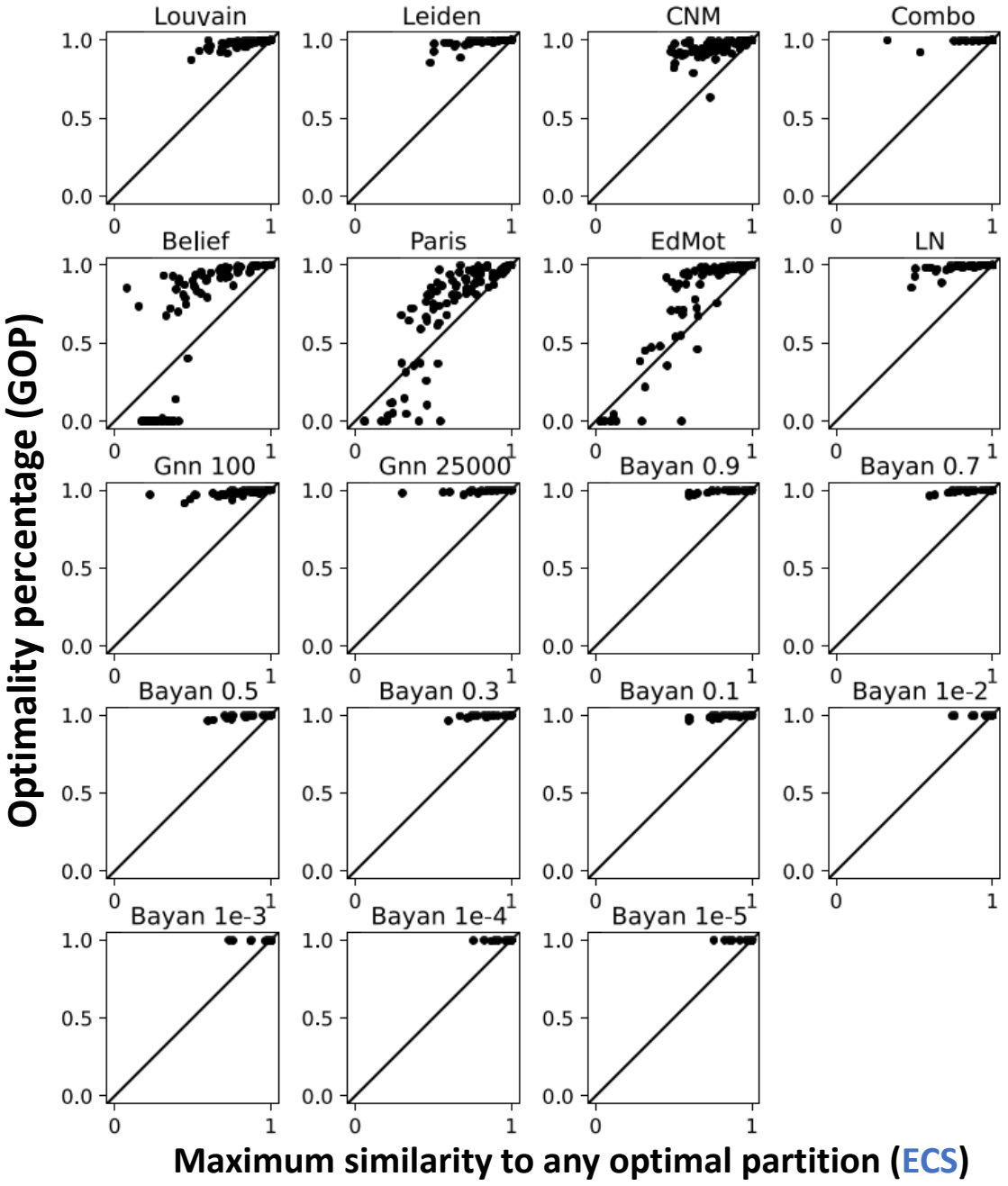
UNIVERSITY OF TORONTO

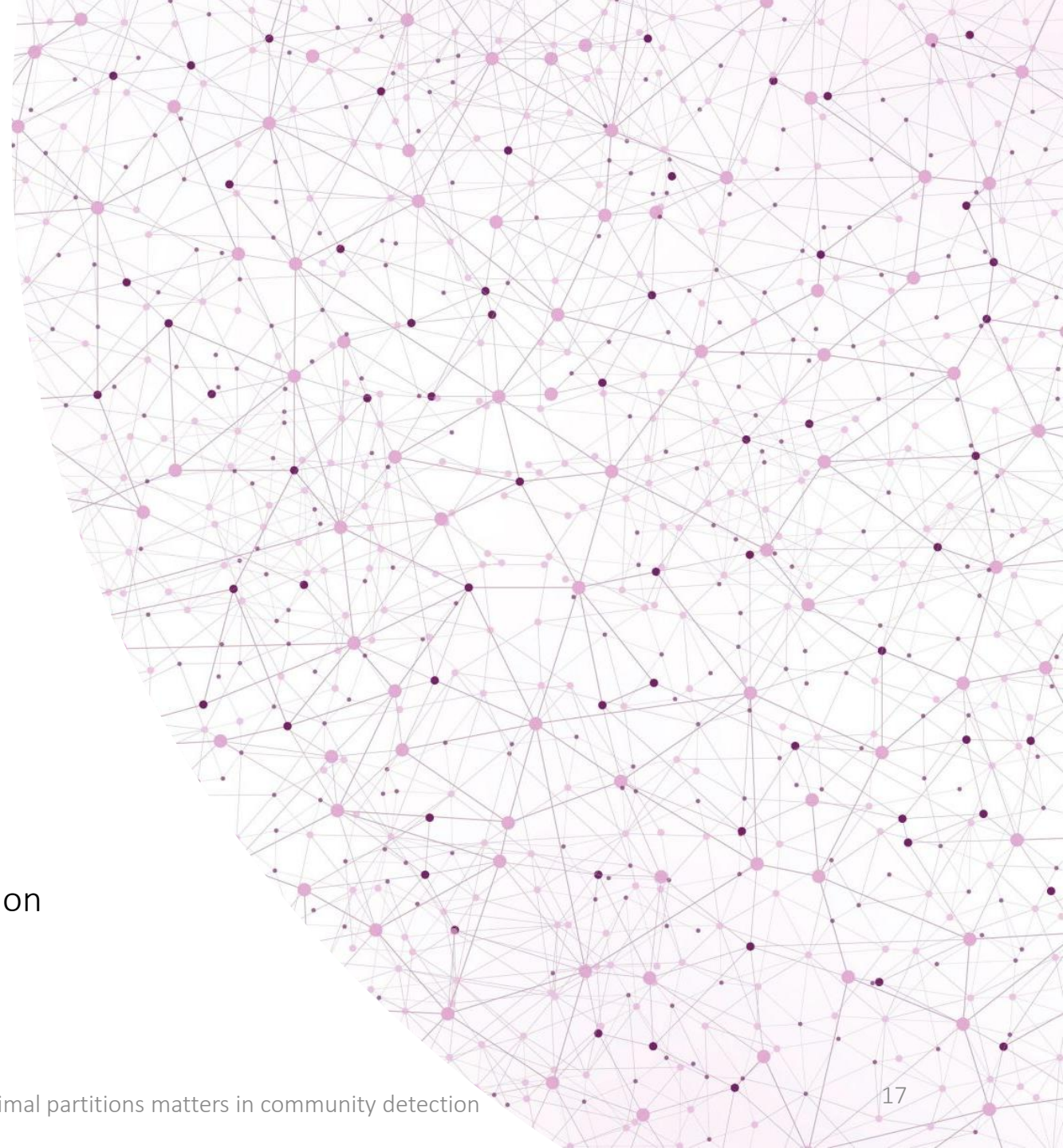
Test cases are 104 graphs with modular structure:

- 54 real networks
- 20 LFR random graphs
- 30 ABCD random graphs

Each datapoint represents the performance of one algorithm on one test case.

1. Y-values: Most partitions obtained from heuristics are sub-optimal
2. X-values: Many partitions are dissimilar to any optimal partition
3. 45°-line: near-optimal partitions are not similar to any optimal partitions

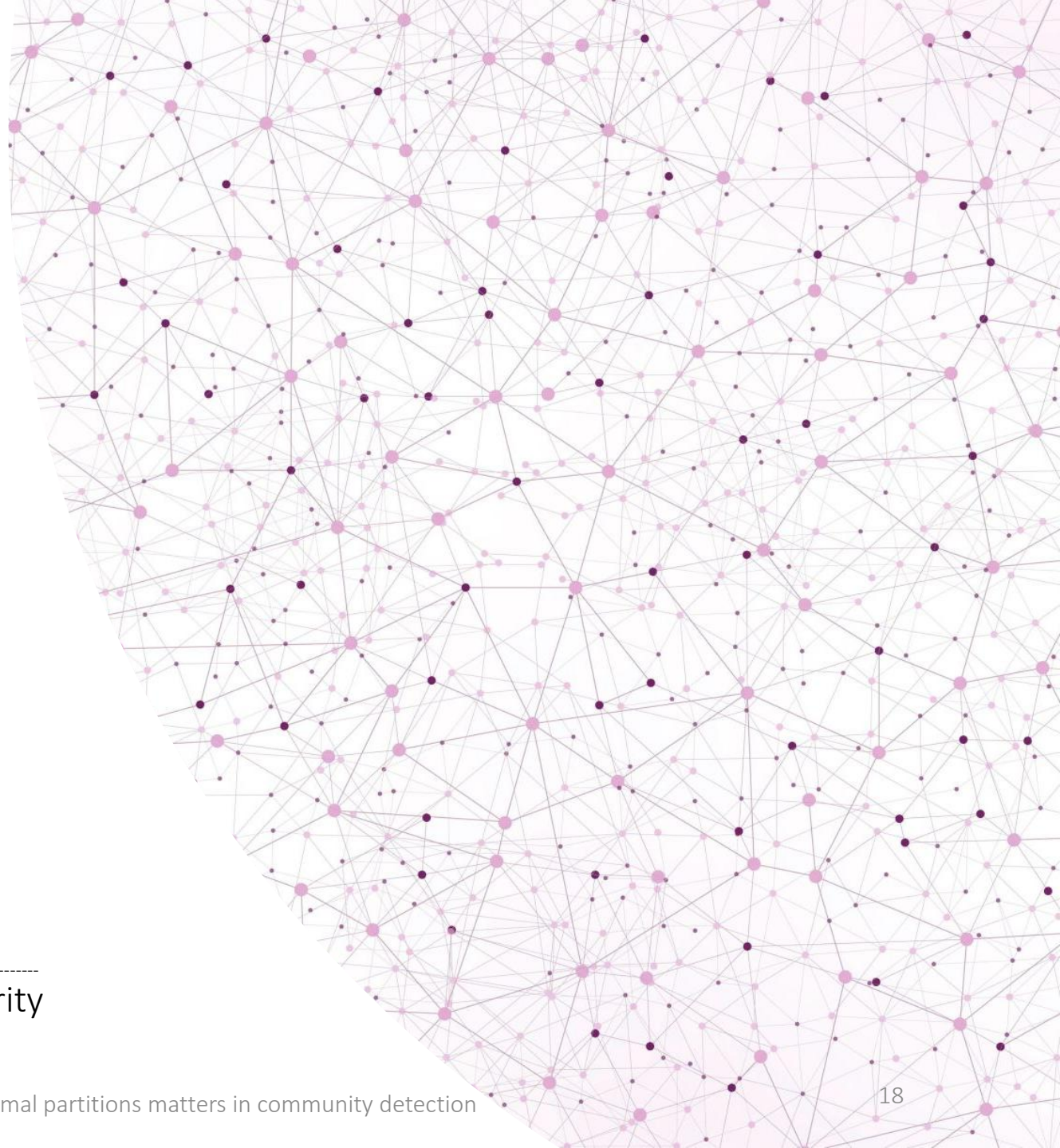




1

Heuristic modularity maximization algorithms rarely* maximize modularity.

*Only 43.9% of the times according to our experiments on 104 synthetic and real networks.



2

Suboptimal partitions of heuristic algorithms are disproportionately dissimilar to any optimal partition.

An x% suboptimality is often associated with a dissimilarity much larger than x% from any optimal partition.

Can we globally maximize modularity?

- YES, for some networks we can!
- Bayan maximizes modularity in networks with fewer than 3000 edges and approximates maximum modularity in slightly larger networks on ordinary computers.

```
%pip install bayanpy
```

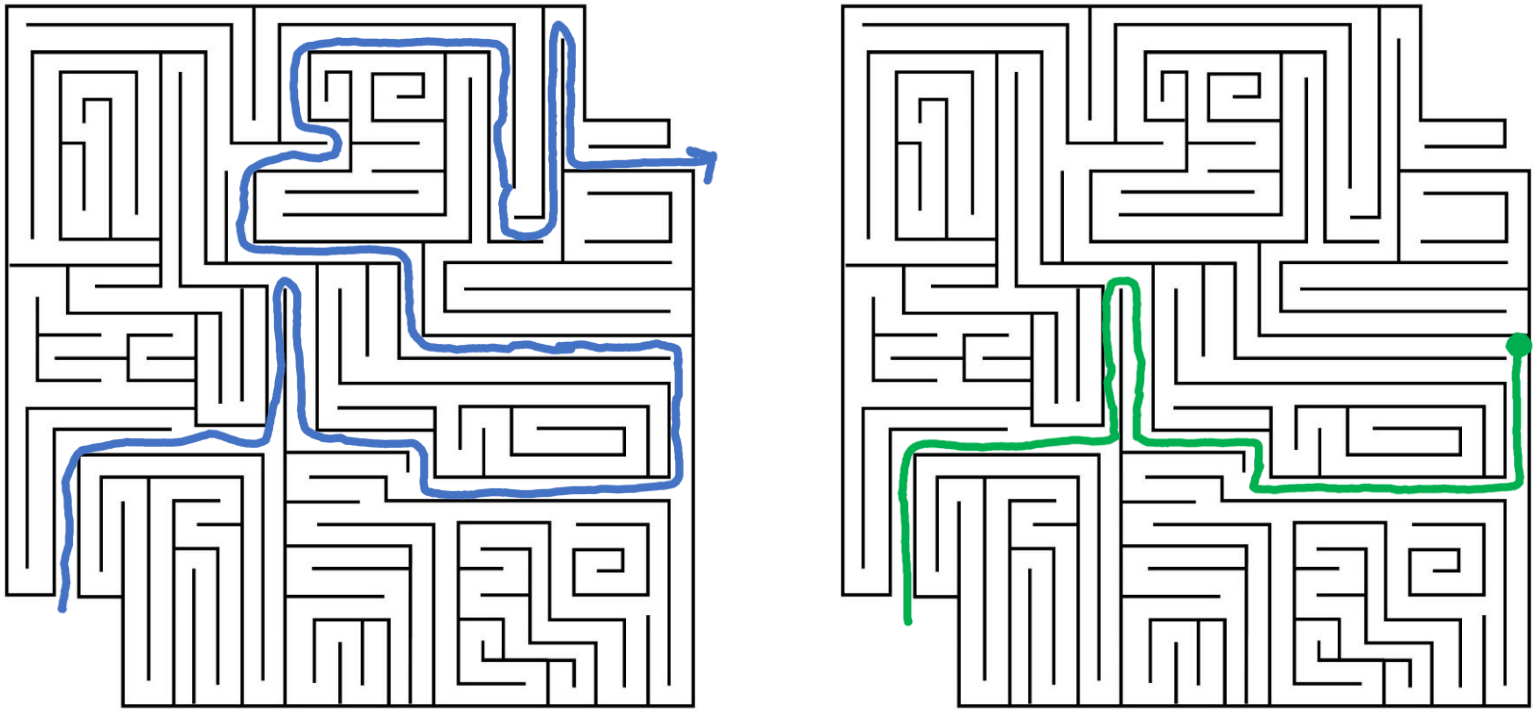
```
import networkx as nx
import bayanpy
```

```
G = nx.barbell_graph(5,2)
```

```
bayanpy.bayan(G)
```

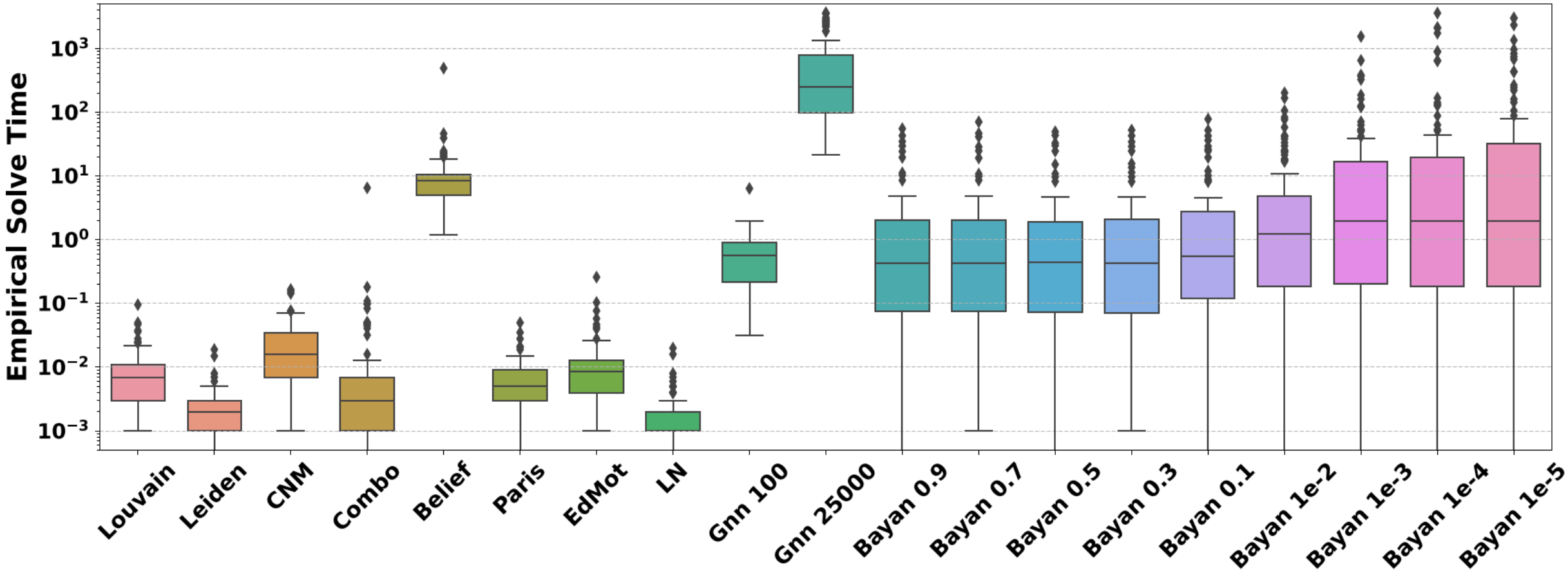
UNIVERSITY OF TORONTO

Assessing the run time of ten modularity-based methods (and their variations)



UNIVERSITY OF TORONTO

Solve time of ten modularity-based methods



How does Bayan work? Solving integer programming by branching on triples

There are 3 constraints for every triple $(i, j, k) \in S \subset V^3$.

All decision variables x_{ij} are binary.

$$x_{ij} + x_{jk} \geq x_{ik}$$
$$x_{ij} + x_{ik} \geq x_{jk}$$
$$x_{ik} + x_{jk} \geq x_{ij}$$

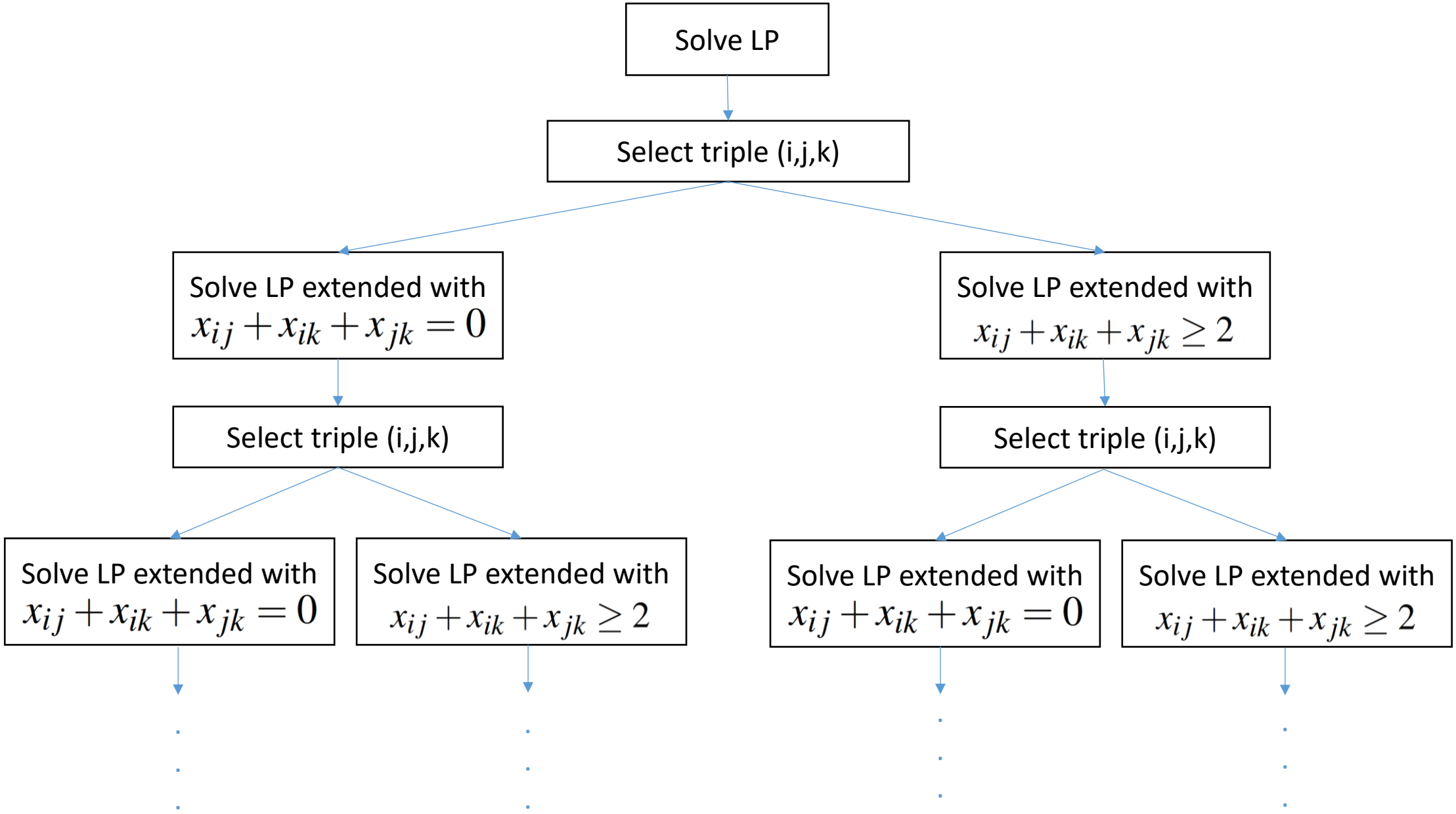
$$x_{ij} + x_{ik} + x_{jk} = 0$$

or

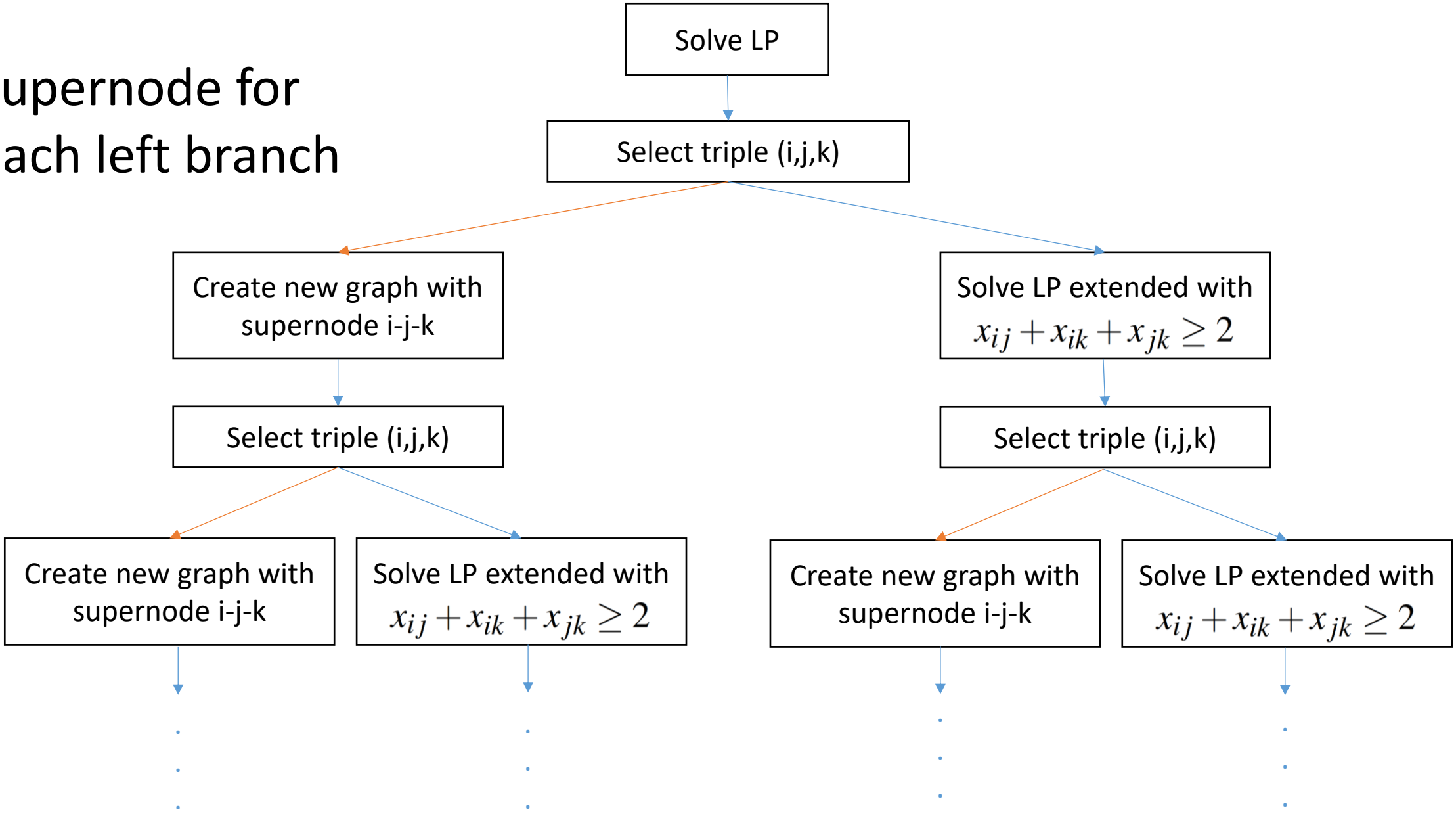
$$x_{ij} + x_{ik} + x_{jk} \geq 2$$

1. Solve the linear programming relaxation
2. Branch on a triple
3. Add a cut
4. Go to 1

UNIVERSITY OF TORONTO


Solve LP
Select triple (i,j,k)
Solve LP extended with $x_{ij} + x_{ik} + x_{jk} = 0$
Solve LP extended with $x_{ij} + x_{ik} + x_{jk} \geq 2$
Select triple (i,j,k)
Select triple (i,j,k)
Solve LP extended with $x_{ij} + x_{ik} + x_{jk} = 0$
Solve LP extended with $x_{ij} + x_{ik} + x_{jk} \geq 2$
Solve LP extended with $x_{ij} + x_{ik} + x_{jk} = 0$
Solve LP extended with $x_{ij} + x_{ik} + x_{jk} \geq 2$

Supernode for each left branch



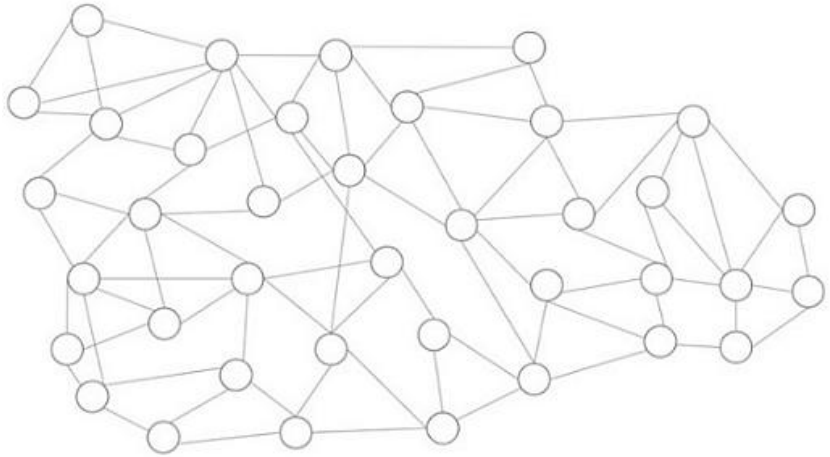
UNIVERSITY OF TORONTO

Wait! But clustering is not about modularity maximization!

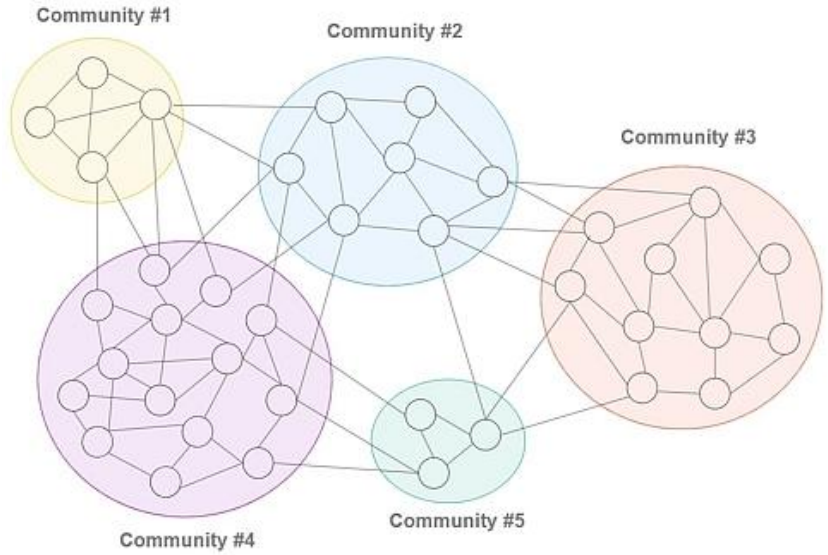
A supervised comparison of 30 CD algorithms

- Input: graph
- Method: a network clustering algorithm
- Output: a partition
- Test: **retrieval of planted partitions** in synthetic benchmark networks (e.g. LFR)

Input

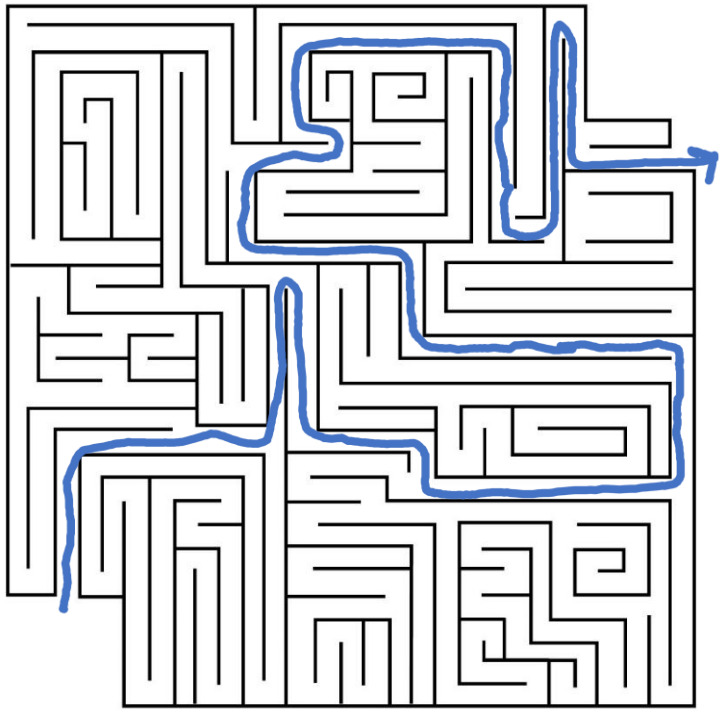


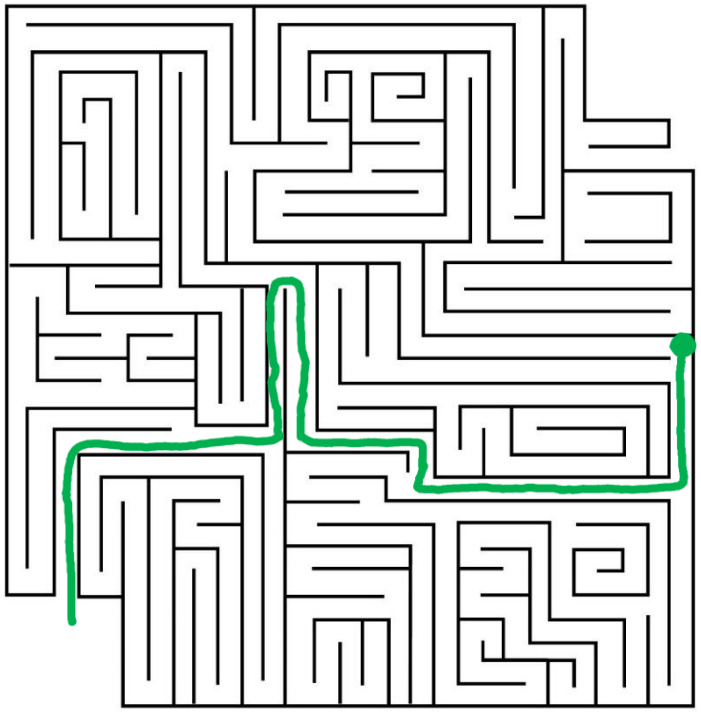
Output

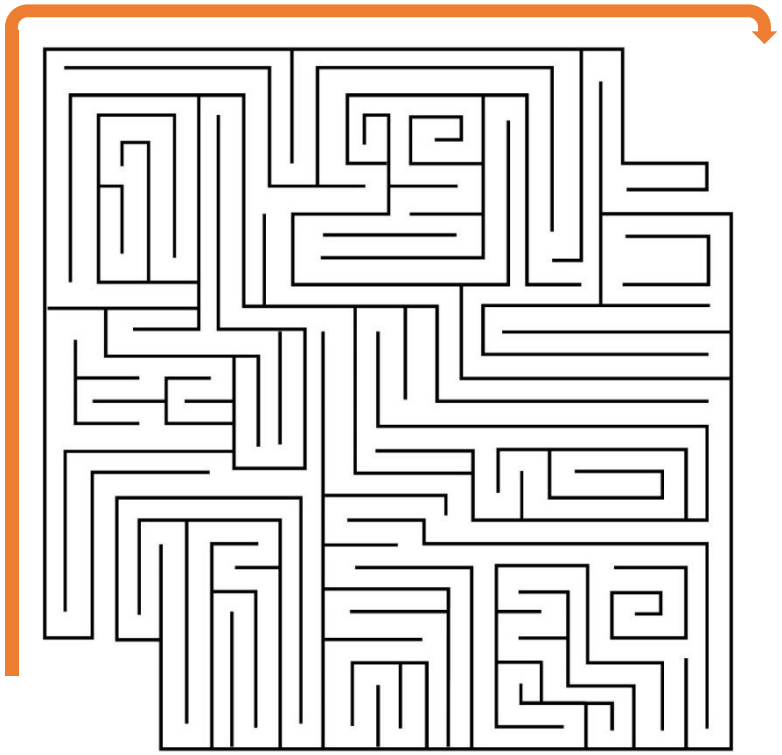


Figures from timbr.ai

Comparing 30 algorithms based on retrieving planted partitions







UNIVERSITY OF TORONTO

Planted partition retrieval assessment

Standard benchmark graphs:

- LFR benchmarks (**L**ancichinetti-**F**ortunato-**R**adicchi benchmarks)
- ABCD benchmarks (**A**rtificial **B**enchmark for **C**ommunity **D**etection)

Retrieval performance indicator:

- Similarity with the planted partitions

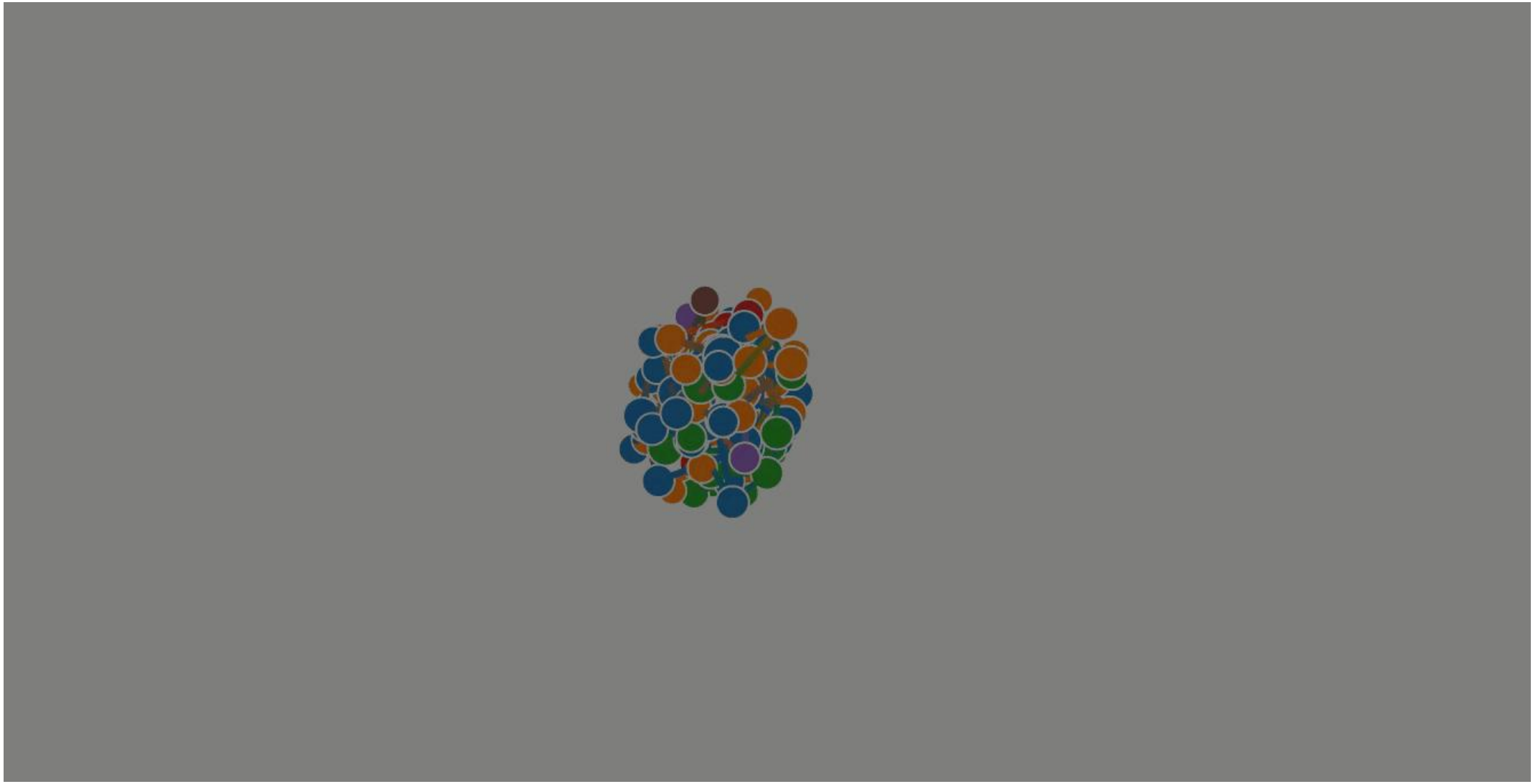
adjusted mutual information (or RMI[1] or ECS[2]) averaged over 100 graphs

1. RMI: Reduced Mutual Information (Newman et al. 2020, Jerdee and Newman 2023)
2. ECS: Element-Centric Similarity (Gates et al. 2019)

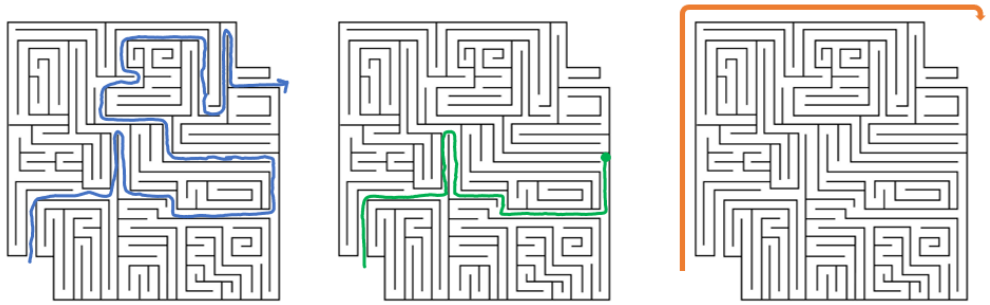
UNIVERSITY OF TORONTO

LFR benchmark with **low** noise

LFR benchmark with **high** noise



Comparing 30 algorithms



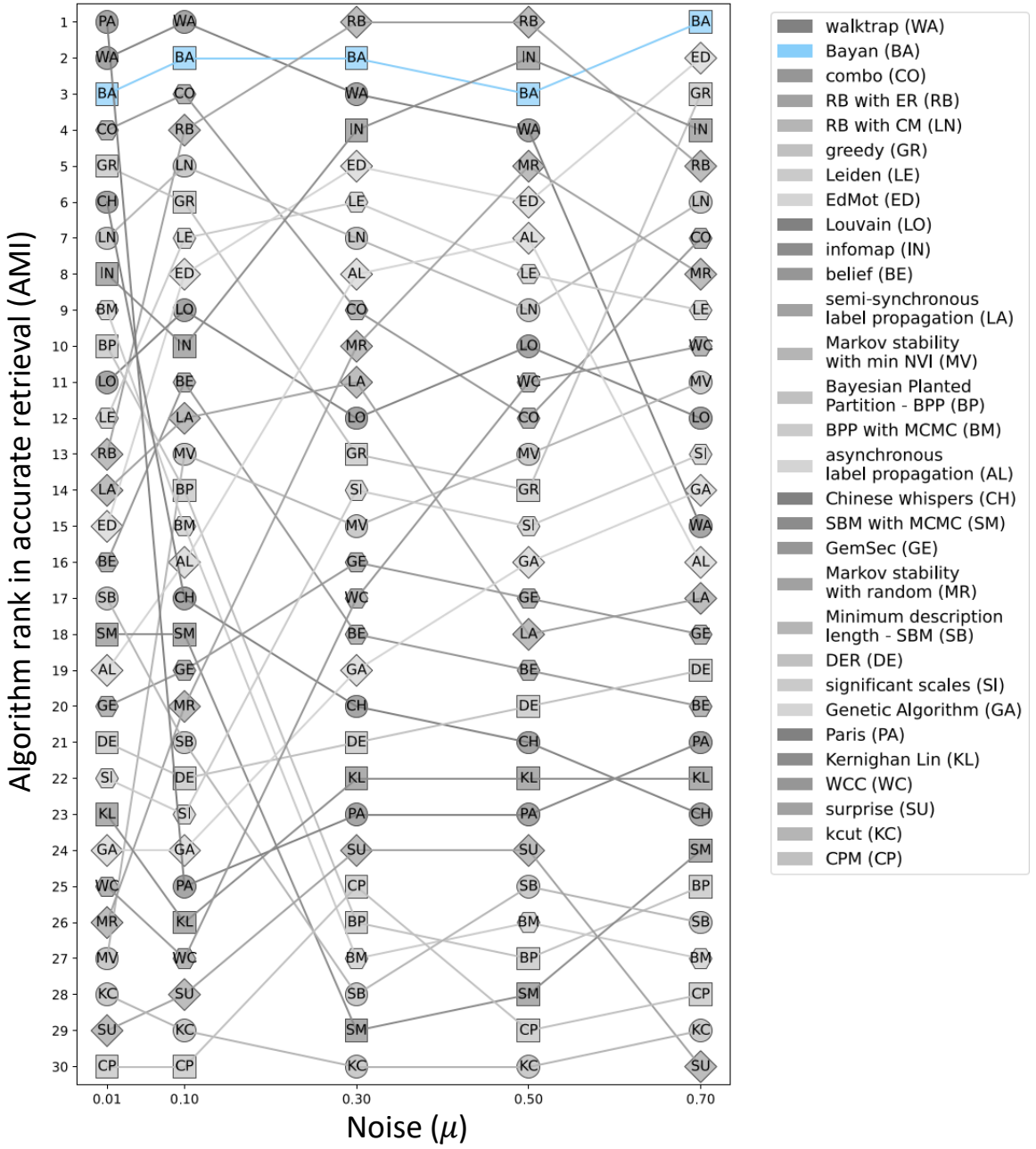
Test cases:

- 500 LFR benchmark graphs with up to 1000 edges (generated with planted partitions)
- Noise μ (fraction of inter-community edges)

$$\mu \in \{1\%, 10\%, 30\%, 50\%, 70\%\}$$

Comparing 30 algorithms

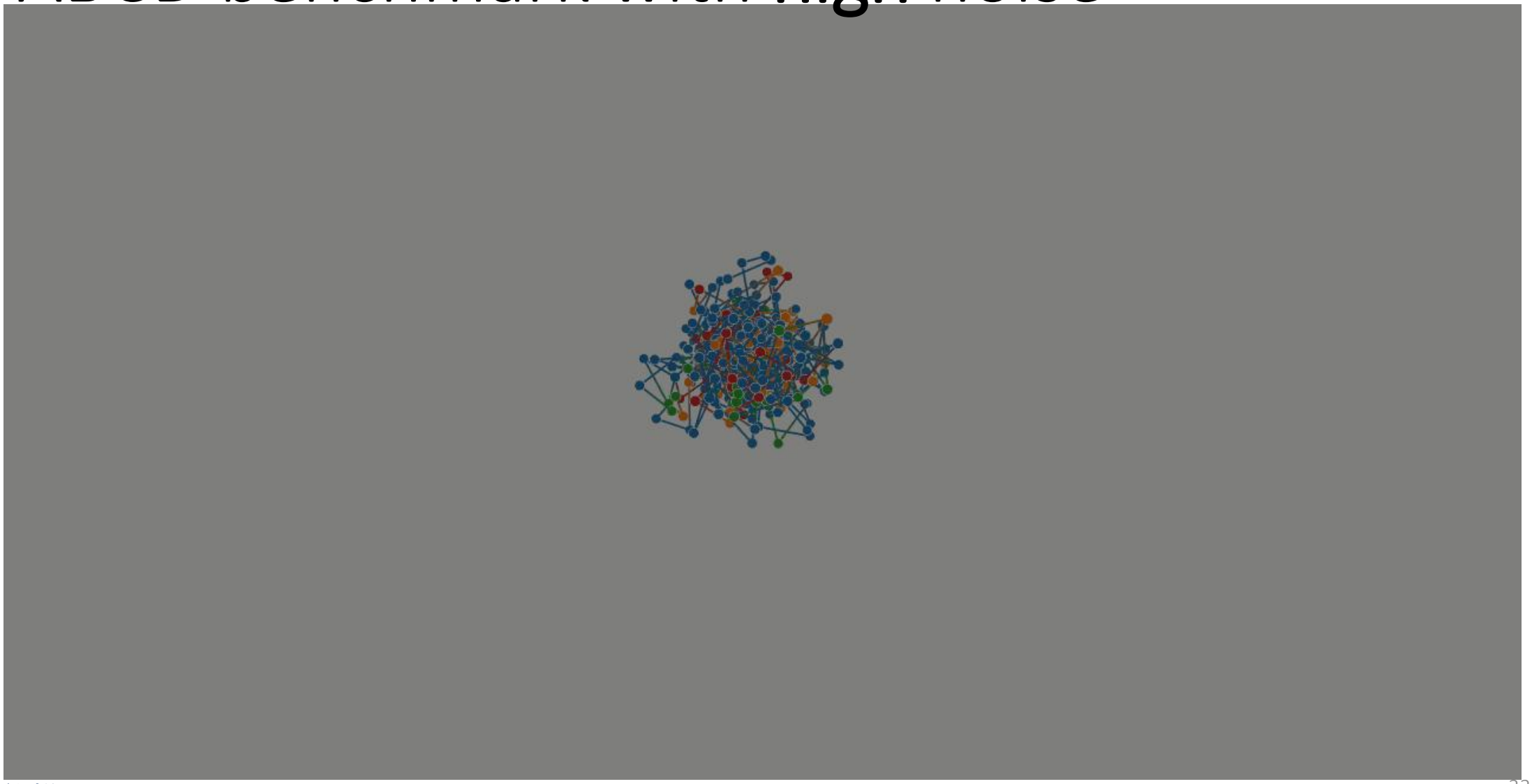
Based on retrieving the planted partitions of LFR benchmarks



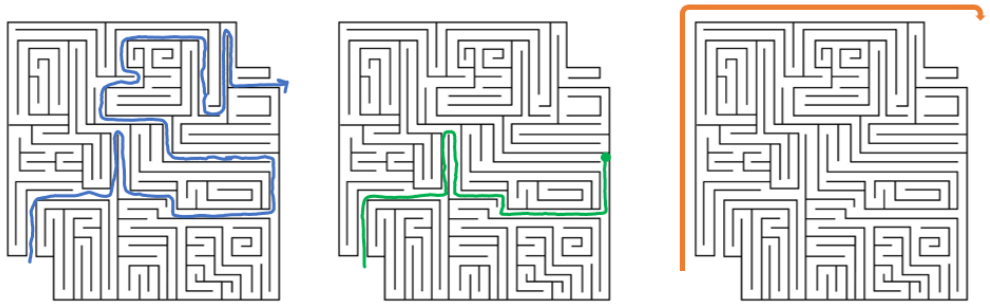
UNIVERSITY OF TORONTO

ABCD benchmark with **low** noise

ABCD benchmark with **high** noise



Comparing 30 algorithms



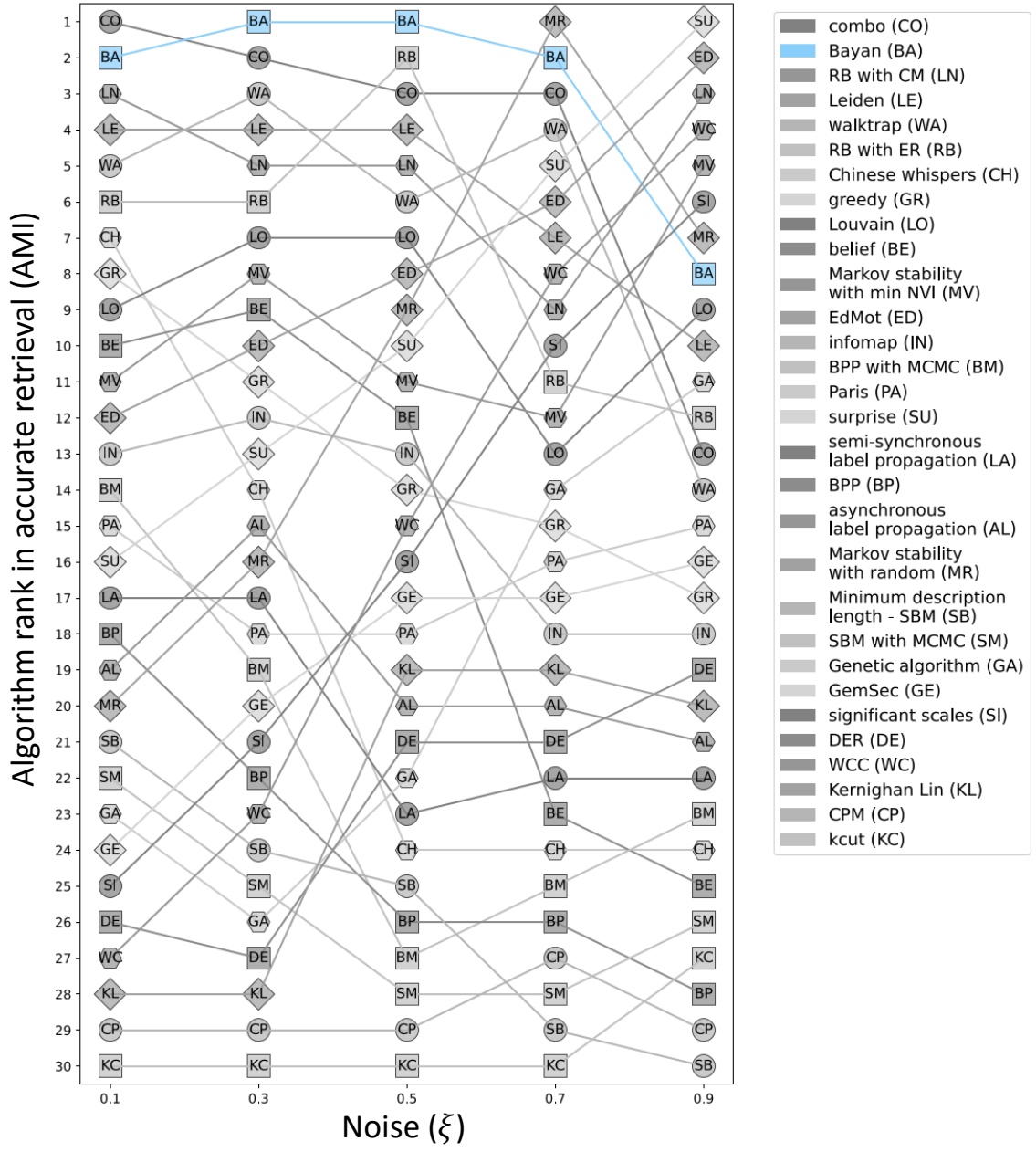
Test cases:

- 500 ABCD benchmark graphs with up to 1000 edges (generated with planted partitions)
- Noise ξ (mixing parameter)

$$\xi \in \{10\%, 30\%, 50\%, 70\%, 90\%\}$$

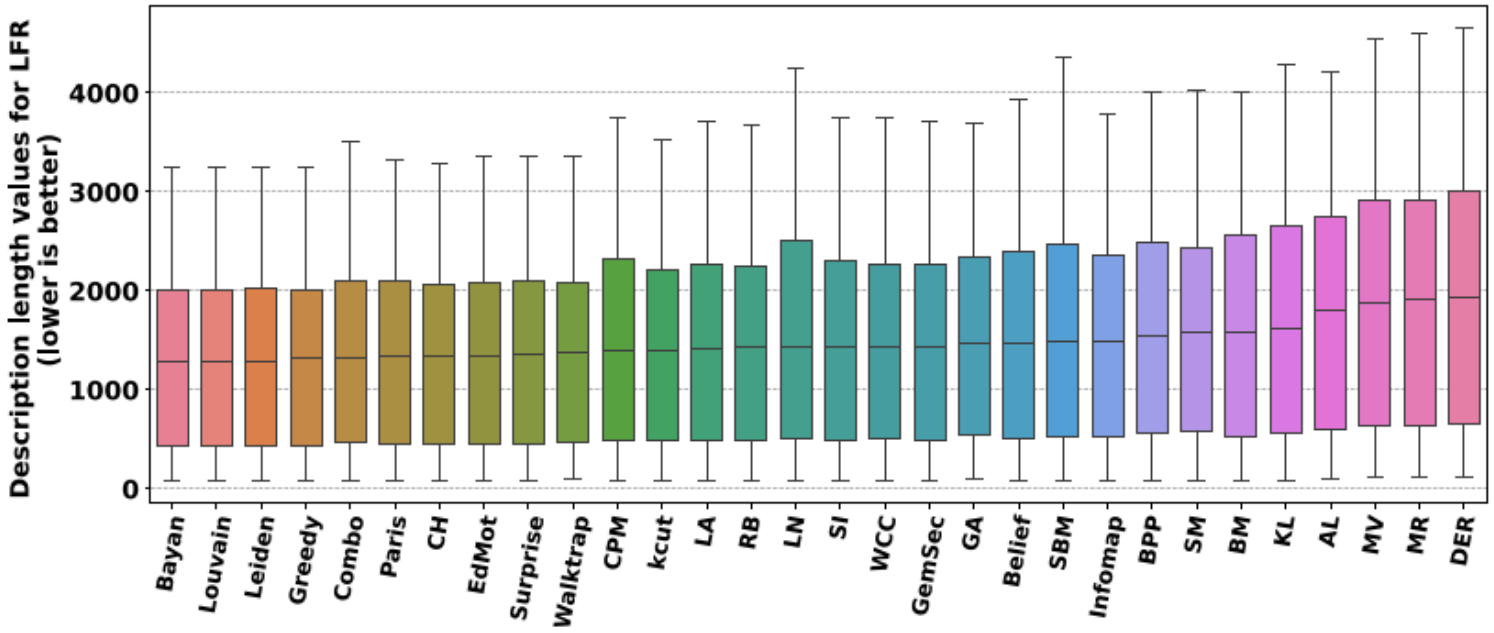
Comparing 30 algorithms

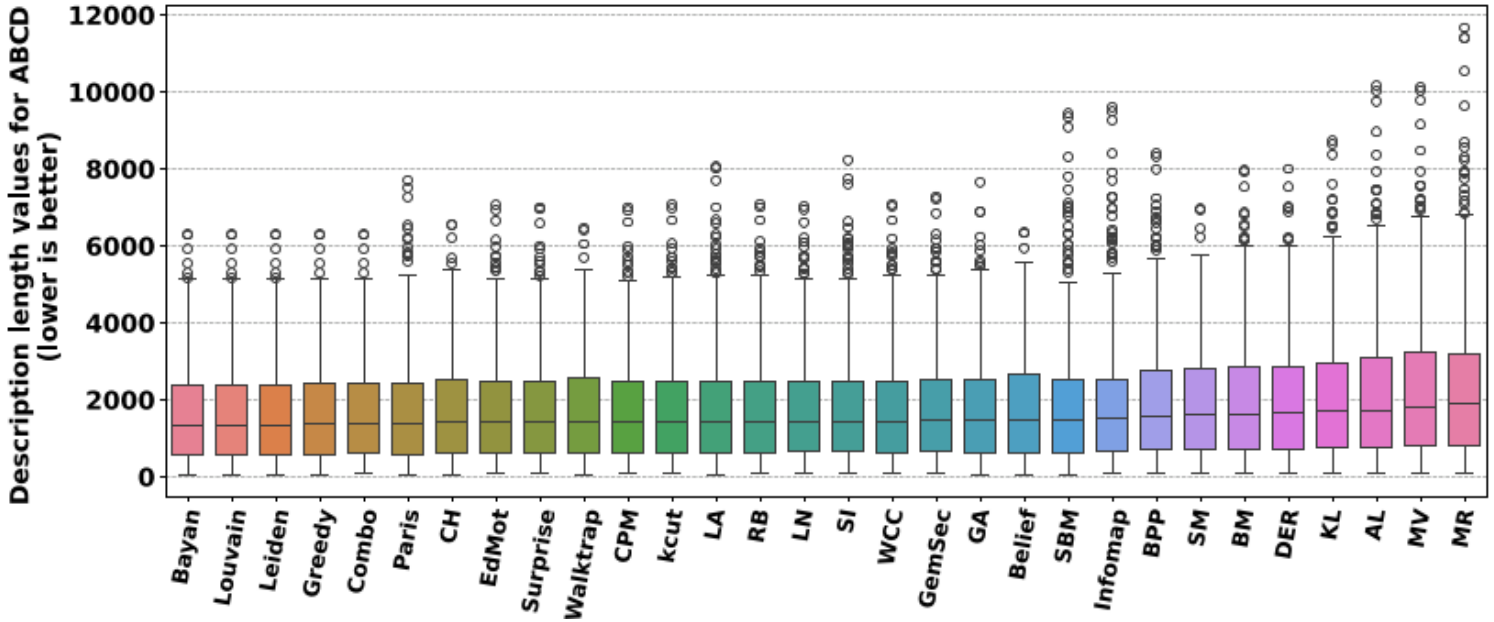
Based on retrieving the planted partition of ABCD benchmark



UNIVERSITY OF TORONTO

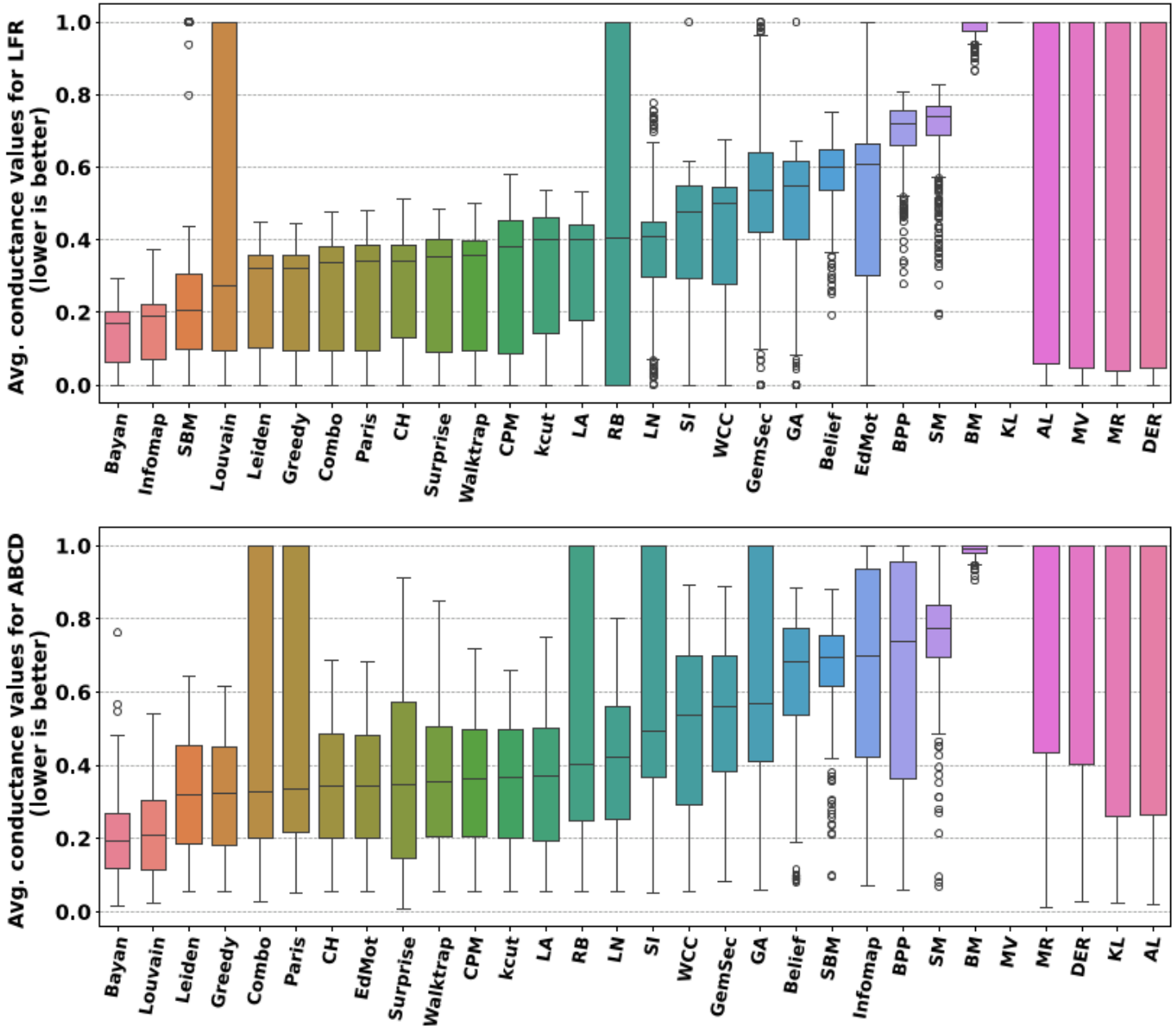
Unsupervised assessment: description length





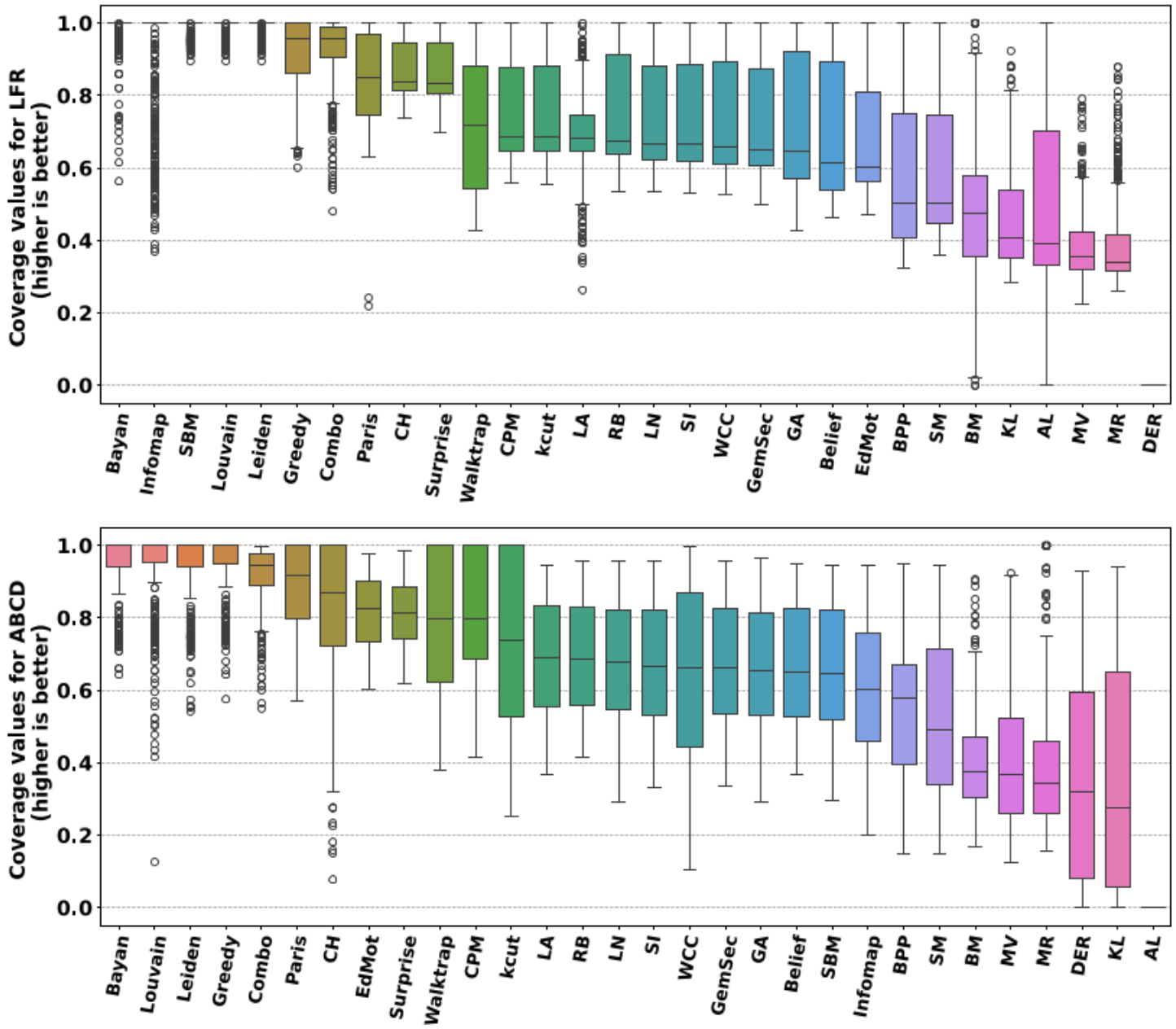
UNIVERSITY OF TORONTO

Unsupervised assessment: average conductance



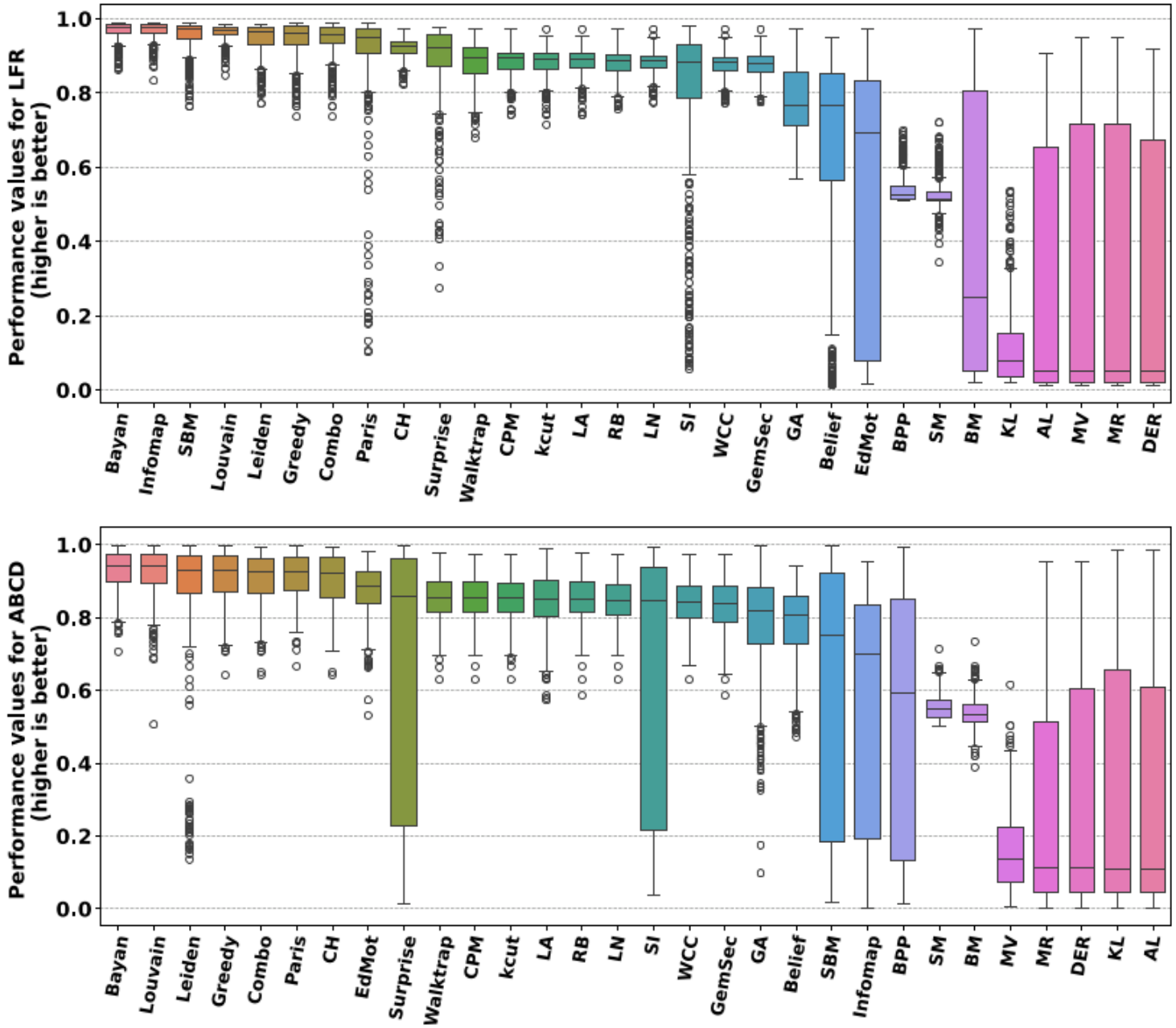
UNIVERSITY OF TORONTO

Unsupervised assessment: partition coverage



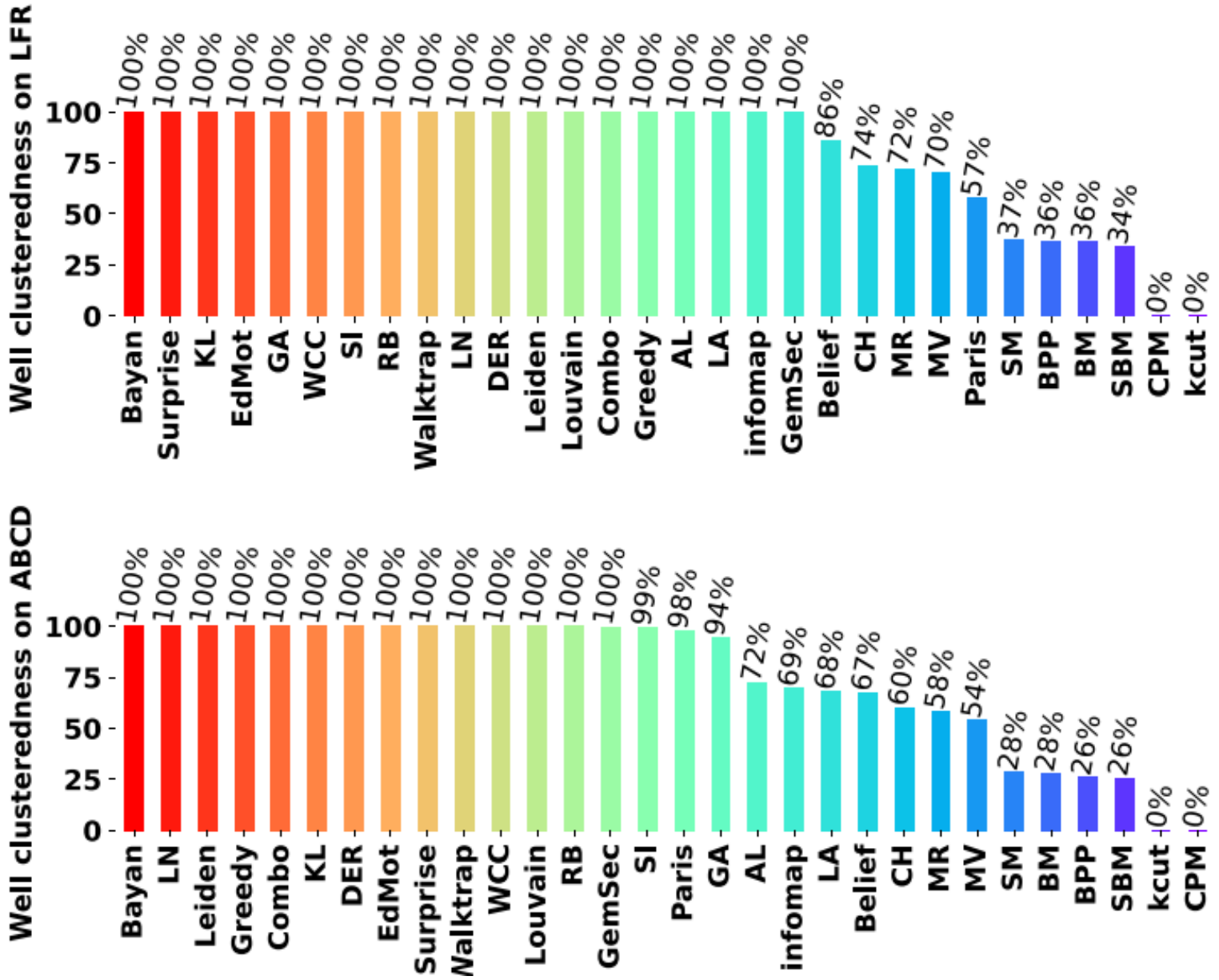
UNIVERSITY OF TORONTO

Unsupervised assessment: partition performance



UNIVERSITY OF TORONTO

Unsupervised assessment: well-clusteredness



UNIVERSITY OF TORONTO

Resources on this project

```
%pip install bayanpy
```

```
import networkx as nx
import bayanpy
```

```
G = nx.barbell_graph(5,2)
```

```
bayanpy.bayan(G)
```

Paper 1: doi.org/10.1007/978-3-031-36027-5_48
Paper 2: doi.org/10.1016/j.jocs.2024.102283
Paper 3: doi.org/10.1103/PhysRevE.110.044315
GitHub Repo: github.com/saref/bayan
Project website: bayanproject.github.io

Google Colab examples:
tinyurl.com/bayancolab

UNIVERSITY OF TORONTO

Special thanks to my co-authors:

Mahdi Mostajabdaveh
Polytechnique Montréal

Hriday Chheda
Department of Mechanical and Industrial Engineering
University of Toronto

UNIVERSITY OF TORONTO

Additional references

Dinh, T.N., Thai, M.T.: Toward optimal community detection: From trees to general weighted networks. Internet Mathematics 11(3), 181–200 (2015)

Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Physical review E 70(6), 066111 (2004)

Biemann, C.: Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In: Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing. TextGraphs-1, pp. 73–80. (2006)

Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. Physical Review E 74(1), 016110 (2006).

Pons, P., Latapy, M.: Computing communities in large networks using random walks. J. Graph Algorithms Appl. 10(2), 191–218 (2006)

Ruan, J., Zhang, W.: An efficient spectral algorithm for network community discovery and its applications to biological and social networks. In: Seventh IEEE International Conference on Data Mining (ICDM 2007), pp. 643–648. IEEE, (2007)

Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment 2008(10), 10008 (2008).

Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences 105(4), 1118–1123 (2008).

Leicht, E.A., Newman, M.E.J.: Community structure in directed networks. Physical Review Letters 100(11), 118703 (2008).

Pizzuti, C.: GA-Net: A Genetic Algorithm for Community Detection in Social Networks. In: Proceedings of the 10th International Conference on Parallel Problem Solving from Nature — PPSN X - Volume 5199, pp. 1081–1090. Springer, Berlin, Heidelberg (2008)

Cordasco, G., Gargano, L.: Community detection via semi-synchronous label propagation algorithms. In: 2010 IEEE International Workshop On: Business Applications of Social Network Analysis (BASNA), pp. 1–8. IEEE, (2010)

Traag, V.A., Van Dooren, P., Nesterov, Y.: Narrow scope for resolution-limit-free community detection. Physical Review E 84(1), 016114 (2011)

Traag, V.A., Krings, G., Van Dooren, P.: Significant scales in community structure. Scientific reports 3(1), 1–10 (2013)

Peixoto, T.P.: Efficient monte carlo and greedy heuristic for the inference of stochastic block models. Physical Review E 89(1), 012804 (2014)

Prat-P´erez, A., Dominguez-Sal, D., Larriba-Pey, J.-L.: High quality, scalable and parallel community detection for large real graphs. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 225–236 (2014)

Sobolevsky, S., Campari, R., Belyi, A., Ratti, C.: General optimization technique for high-quality community detection in complex networks. Physical Review E 90(1), 012811 (2014)

Zhang, P., Moore, C.: Scalable detection of statistically significant communities and hierarchies, using message passing for modularity. Proceedings of the National Academy of Sciences 111(51), 18144–18149 (2014)

Traag, V.A., Aldecoa, R., Delvenne, J.-C.: Detecting communities using asymptotical surprise. Physical Review E 92(2), 022816 (2015)

Bonald, T., Charpentier, B., Galland, A., Hollocou, A.: Hierarchical graph clustering using node pair sampling. In: MLG 2018 - 14th International Workshop on Mining and Learning with Graphs, London, UK (2018)

Traag, V.A., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports 9(1) (2019).

Li, P.-Z., Huang, L., Wang, C.-D., Lai, J.-H.: EdMot: An edge enhancement approach for motif-aware community detection. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 479–487 (2019)

Rozemberczki, B., Davies, R., Sarkar, R., Sutton, C.: Gemsec: Graph embedding with self clustering. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 65–72 (2019)

UNIVERSITY OF TORONTO

1. The average modularity-based heuristic yields optimal partitions for only 43.9% of the 104 networks analyzed.

2. Suboptimal partitions of heuristic algorithms are disproportionately dissimilar to any optimal partition.

3. Bayan maximizes modularity in mid-sized networks and approximates maximum modularity in slightly larger networks on ordinary computers.

4. Modularity-maximum partitions retrieve planted partitions of LFR and ABCD graphs at rates higher than most other algorithms.

See Bayan on your smartphone ->

Thank you!

Questions?

UNIVERSITY OF TORONTO