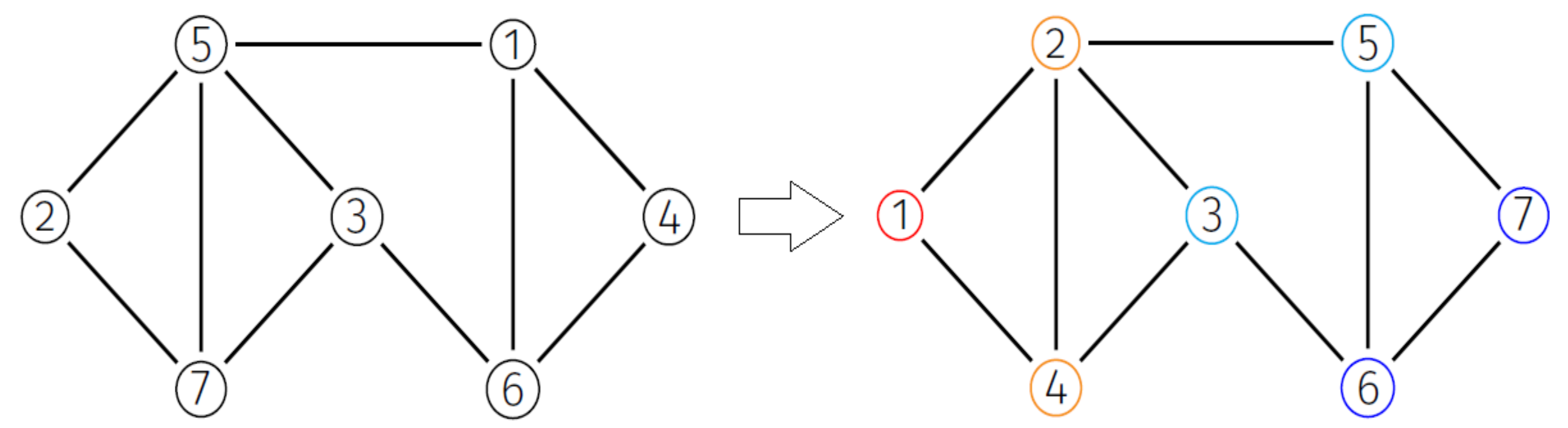
Evaluation of Vertex Reordering for Graph Applications

Reet Barik, PhD Candidate, School of EECS, Washington State University
Advisor: Ananth Kalyanaraman (WSU), Co-Advisor: Mahantesh Halappanavar (PNNL)

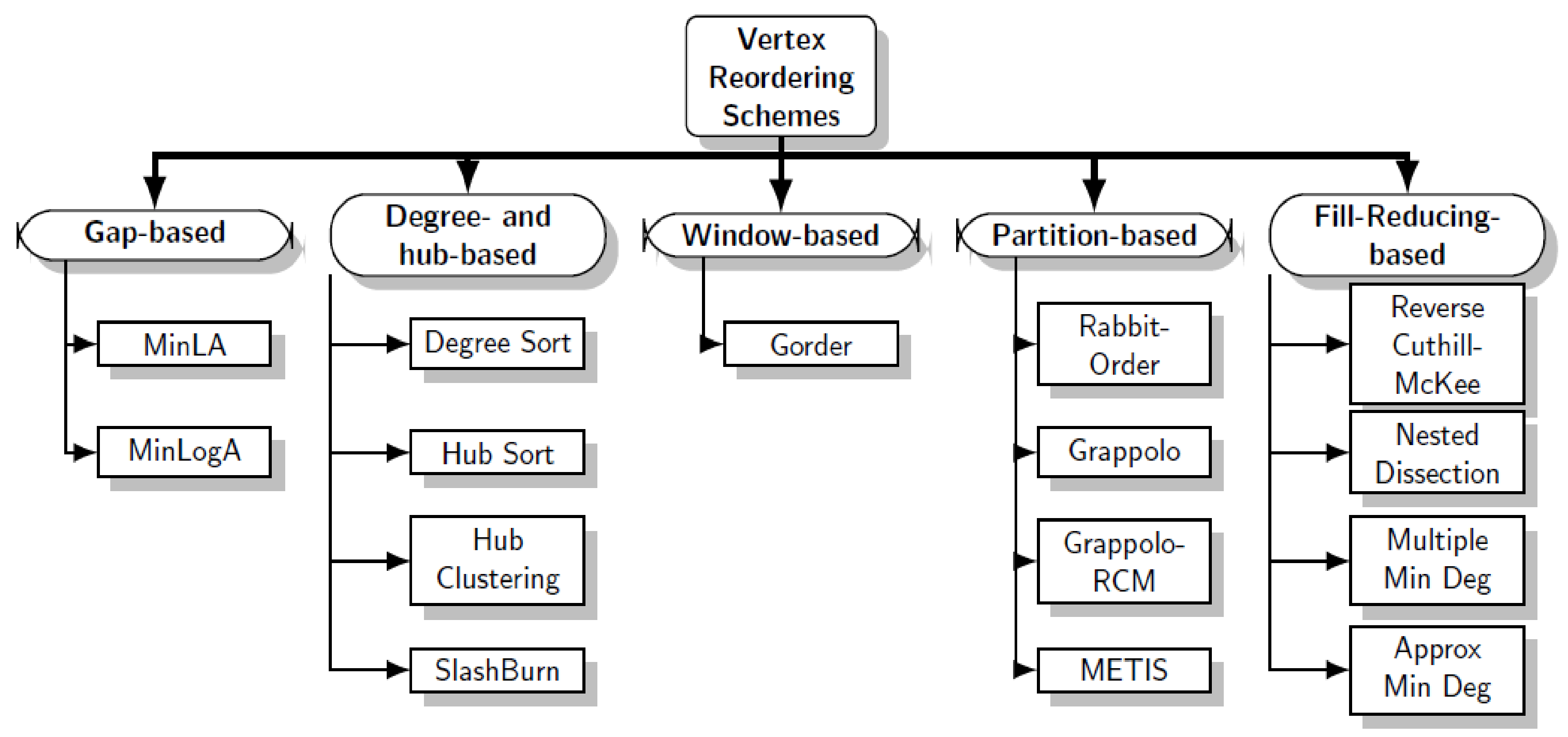
Vertex Reordering

❑ **Vertex Reordering:** Let $G = (V, E)$ be an input graph, with V being the set of (n) vertices indexed continuously starting from 1 till n and E being the set of (m) edges.

A *vertex ordering* Π of V is a 1-1 mapping (bijection or permutation) of V onto a sequence or linear order.



❑ **Classification:** Different reordering schemes, categorized below, use different objectives or measures to achieve their goal.



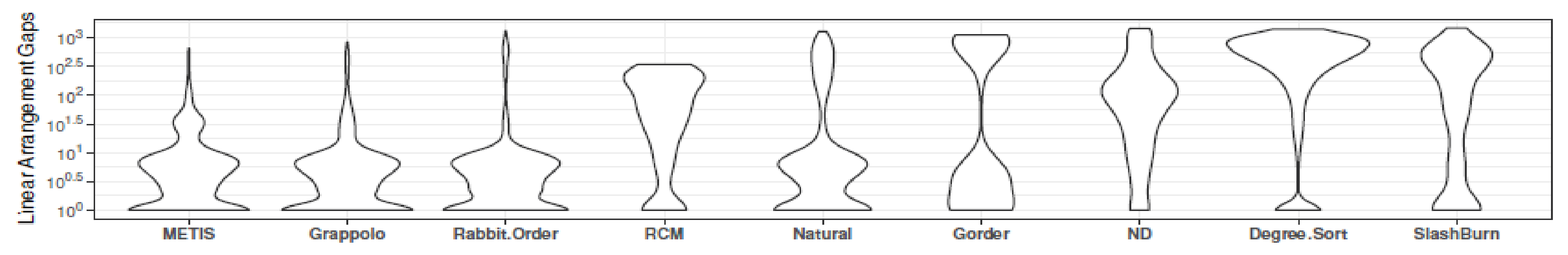
❑ **Comparative Evaluation Metric:** The *average linear arrangement gap* is a metric defined in [1] to quantifying the efficacy of a reordering scheme in preserving locality. Given an ordering Π of V, for two vertices $i, j \in V$, connected by an edge, the gap in Π between vertices *i* and *j* is given by:

$$\xi_{\Pi}(i,j) = |\Pi(i) - \Pi(j)|$$

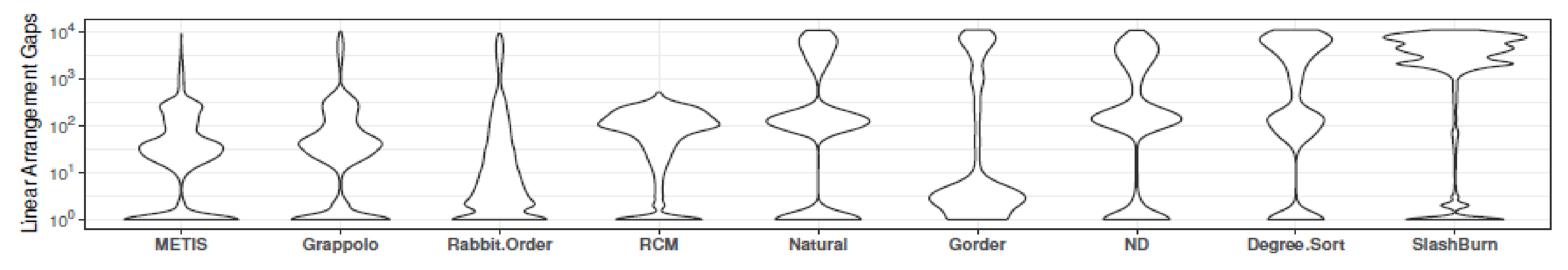
The *average linear arrangement gap* is then defined as:

$$\hat{\xi}(G,\Pi) = \frac{1}{|E|} \sum_{(i,j)\in E} \xi_{\Pi}(i,j)$$

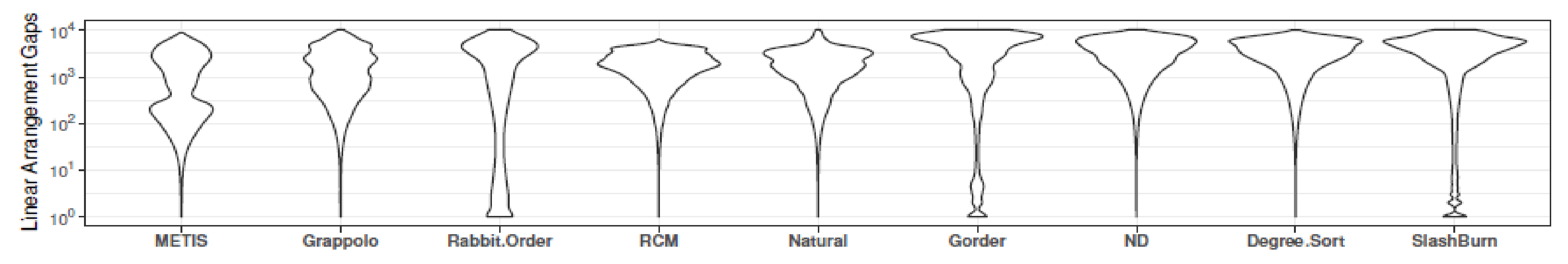
Evaluating Gap Measures



$\hat{\xi}$:	12.17	14.04	21.24	99.10	84.45	304.26	202.16	502.44	249.76
β:	648.00	847.00	1,308.00	343.00	1,278.00	1,120.00	1,437.00	1,401.00	1,466.00
$\hat{\beta}$:	4.79	8.85	13.46	87.68	55.70	57.03	106.94	443.63	22.32



$\hat{\xi}$:	124.03	229.59	217.41	103.33	1,447.11	1,350.39	1,379.34	2,591.26	4,023.82
β:	8,988.00	10,193.00	9,196.00	516.00	10,838.00	11,076.00	10,938.00	11,102.00	11,138.00
$\hat{\beta}$:	172.52	325.84	314.99	146.43	1,346.06	2,139.75	1,903.20	3,361.55	2,501.23



$\hat{\xi}$:	1,392.10	1,964.45	2,323.13	1,876.73	1,986.26	3,872.86	3,528.15	3,439.97	3,786.74
β:	9,015.00	10,369.00	10,457.00	6,405.00	10,386.00	10,456.00	10,433.00	10,288.00	10,495.00
$\hat{\beta}$:	1,942.46	2,684.47	2,751.34	1,960.51	1,846.72	2,359.62	3,974.69	4,839.00	1,328.19

Violin plots of the gap distribution and gap metrics for different methods (blue is best, and red is worst) for input graphs Chicago, Fe4elt2, and vsp (shown in clockwise order from top left).

Key Experimental Results

❑ **Qualitative Results:** Figure 1. shows how different schemes compare based on the average gap profile:

- METIS, Grappolo, and Rabbit-Order (in red) outperform (5x-10x for at least 75% of the datasets) others.
- RCM (black) shows competitive performance as well.

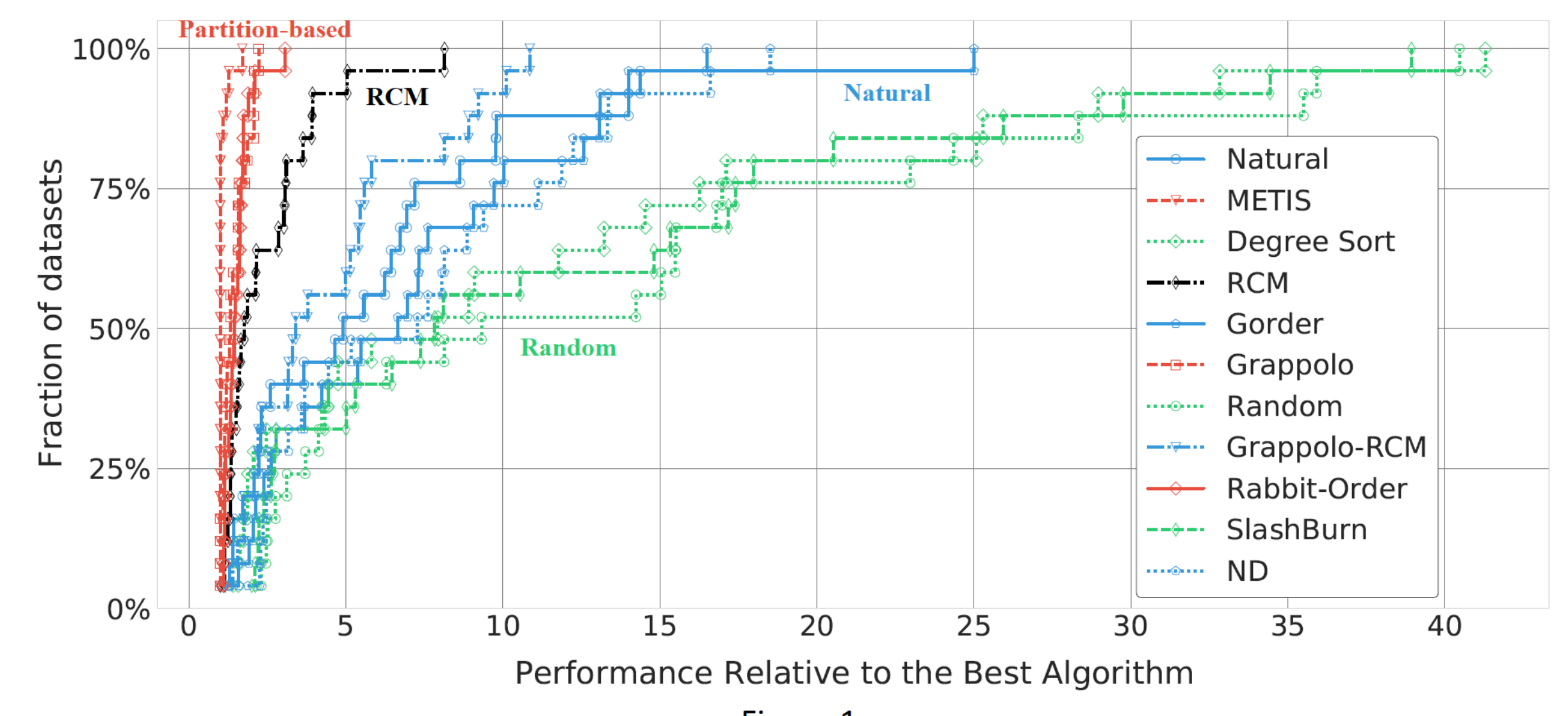


Figure 1.

	Phase (s)				Iteration (s)				Iteration Count				Modularity (final)				Work% (CPU Time)				Work/edge (loads)			
Graph	Grappolo	RCM	Natural	Degree	Grplo	RCM	Ntrl	Degr	Grplo	RCM	Ntrl	Degr	Grplo	RCM	Ntrl	Degr	Grplo	RCM	Ntrl	Degr	Grplo	RCM	Ntrl	Degr
LiveMocha	1.4	1.0	1.5	1.5	0.20	0.26	0.37	0.51	7	4	4	3	0.040	0.047	0.027	0.019	28%	24%	20%	19%	3.1	3.2	3.8	2.8
CA RoadNet	0.9	1.5	0.9	1.9	0.13	0.38	0.23	0.65	7	4	4	3	0.992	0.992	0.992	0.992	9%	12%	9%	16%	0.5	1.5	0.8	2.5
Hyves	1.6	1.5	1.7	2.4	0.39	0.39	0.44	0.60	4	4	4	4	0.608	0.731	0.722	0.715	22%	19%	18%	17%	2.5	2.5	2.1	2.1
arXiv hep-ph	3.3	3.8	4.2	5.8	0.09	0.10	0.12	0.14	36	37	36	42	0.516	0.530	0.531	0.535	50%	45%	39%	36%	0.9	1.0	0.9	0.9
YouTube	4.2	3.4	9.5	7.3	0.28	0.85	0.48	1.82	15	4	20	4	0.644	0.636	0.644	0.633	21%	12%	19%	13%	6.4	7.1	6.0	7.7
Skitter	6.3	2.9	8.1	5.8	0.14	0.95	0.21	1.45	44	3	39	4	0.842	0.833	0.840	0.827	20%	11%	15%	12%	3.2	7.0	3.4	7.3
Actor collab	7.7	12.4	8.5	29.2	0.16	0.26	0.21	0.58	49	48	41	50	0.708	0.715	0.714	0.717	32%	24%	27%	20%	2.3	2.3	2.4	2.5
LiveJournal	24.0	49.5	52.3	90.6	0.25	0.92	0.66	1.41	96	54	79	64	0.746	0.751	0.749	0.741	35%	29%	27%	30%	9.9	10.9	10.4	11.7
Orkut	70.1	131.7	52.9	131.1	0.55	1.25	0.62	2.11	128	105	85	62	0.623	0.622	0.635	0.623	38%	44%	38%	41%	8.8	10.9	9.4	11.9
	lower (redder) better				lower (redder) better				lower (redder) better				higher (redder) better				higher (redder) better				lower (redder) better			

Figure 2.

❑ **Application Impact:** Figure 2. compares the effect of four schemes on a parallel community detection application:

- Grappolo usually outperforms Degree Sort by factors 2x–4x when looking at Phase and Iteration times.
- Grappolo ordering also usually has the highest Parallel Efficiency (Work%).
- Grappolo ordering has the lowest work per edge which results in a better load balance.

Note: Little separation was observed among different reordering schemes for Influence Maximization due to the numerous parallel probabilistic BFS's that characterizes its core routine.

Acknowledgments

❑ I would like to thank my co-authors Marco Minutoli and Nathan R. Tallent for their efforts.
❑ This research is in parts supported by the Exascale Computing Project (17-SC-20-SC) through the ExaGraph project at the Pacific Northwest National Laboratory (PNNL); by the U.S. National Science Foundation (NSF) grants CCF 1815467, OAC 1910213, and CCF 1919122 to Washington State University.

References

1. Reet Barik, Marco Minutoli, Mahantesh Halappanavar, Nathan R Tallent, and Ananth Kalyanaraman. Vertex reordering for real-world graphs and applications: An empirical evaluation. In 2020 IEEE International Symposium on Workload Characterization (IISWC), pages 240–251. IEEE, 2020.