Technical Report for "Anchored Maximum Communities over Large Directed Graphs"

THEORETICAL ANALYSIS

Theorem 1. *ADCM is NP-hard for any $k \geq 1$ or $l \geq 1$.*

Proof. We prove NP-hardness by reducing Max-3SAT [4] to ADCM.

Max-3SAT takes a 3-CNF formula ϕ with m clauses and n Boolean variables; the goal is to find a truth assignment to the variables that maximizes the number of satisfied clauses s.

Graph Construction (G, b, k, l). Given ϕ, we construct a directed graph $G = (V, E)$ in polynomial time.

Case 1: $k \geq 1$ and $l \geq 0$.

1. Vertex Set and Parameters. $V = V_L \cup V_C \cup C_{aux}$, where $|V_L| = 2n$ and $|V_C| = m$. Set Auxiliary Core Size $\boldsymbol{|C_{aux}| = \max(k, l) + 1}$, and Anchor Budget $\boldsymbol{b = n}$. The $b = n$ ensures the optimal anchor set $\boldsymbol{A^*}$ encodes a full Boolean assignment.

2. Edge Set E. *(i) $\boldsymbol{E_{aux}}$ (Auxiliary Core).* C_{aux} forms a directed complete clique, guaranteeing $\boldsymbol{C_{aux} \subseteq S}$. *(ii) $\boldsymbol{E_{CA}}$ (Peeling Resistance).* For each $c_j \in V_C$, add l edges $u \to c_j$ ($u \in C_{aux}$) for in-degree guarantee, and $k - 1$ edges $c_j \to u$ ($u \in C_{aux}$) for the out-degree gap. *(iii) $\boldsymbol{E_{CL}}$ (Activation).* For every literal ℓ in C_j, add one edge $c_j \to \ell$.

Constraint: Vertices in V_L have no outgoing edges to any vertex outside A^*.

3. Core Analysis. Let A^* be an optimal anchor set and $\mathrm{S} = \mathbf{C_{k,l}}(\mathbf{G_{A^*}})$ the resulting anchored core.

(i) Since $\ell \in V_L \setminus A^*$ has $\boldsymbol{\delta^{out}(\ell) = 0}$, all non-anchored literal vertices are removed (as $k \geq 1$). Thus, S consists of C_{aux} and a subset of V_C. (ii) The in-degree constraint $\delta^{in}(c_j) \geq l$ is guaranteed by E_{CA} edges from $C_{aux} \subseteq S$.

Claim. $c_j \in S \Leftrightarrow C_j$ is satisfied by A^*.

Proof. ($\Leftarrow$): If C_j is satisfied, $\exists \ell \in C_j$ with $\ell \in A^*$. Then $c_j \to \ell$ ensures $\delta^{out}(c_j, S) \geq (k - 1) + 1 = k$. Since $\delta^{in}(c_j, S) \geq l$ is met by E_{CA} edges, $c_j \in S$. ($\Rightarrow$): If $c_j \in S$, $d_{\text{out}}(c_j, S \cup A^*) \geq k$. As E_{CA} gives only $k - 1$, an E_{CL} edge $c_j \to \ell$ must reach $\ell \in A^*$, so C_j is satisfied. (iii) Objective Equivalence. The ADCM goal is to maximize $|F(A, G)|$. By the Claim, $S = C_{k,l}(G_{A^*})$ has size $|S| = |C_{aux}| + s$, and $F(A^*, G) = S \setminus (C_{k,l}(G) \cup A^*)$, thus $|F(A^*, G)| = s$. Maximizing $|F(A, G)|$ equals maximizing s. As Max-3SAT is NP-hard, ADCM is NP-hard for all $k \geq 1$ and $l \geq 0$.

Case 2: $k \geq 0$ and $l \geq 1$. By symmetry, the result follows from Case 1 applied to the reverse graph G^R. □

Theorem 2. *The ADCM problem cannot be approximated in polynomial time within a factor of $\boldsymbol{\alpha} + \varepsilon$ for any constant $\varepsilon > 0$, unless* P = NP.

Proof. We prove the inapproximability of ADCM via a reduction from the gap version of Max-3SAT.

Gap Setup. We use Håstad's celebrated result [3] on the optimal inapproximability of Max-3SAT. Specifically, for every constant $\varepsilon' > 0$, it is NP-hard to distinguish between instances where all clauses can be satisfied (YES-instances) and instances where at most a fraction $\alpha + \varepsilon'$ of clauses can be satisfied (NO-instances), with

$$\boldsymbol{\alpha = 7/8}.$$

This α represents the best possible polynomial-time approximation ratio for Max-3SAT unless P = NP. For simplicity, we set $\varepsilon' = \varepsilon/2$.

Reduction and Objective Equivalence. Applying the polynomial-time reduction from Theorem 1 to such an instance ϕ (with m clauses) yields an ADCM instance (G, b, k, l) with

$$OPT_{\text{ADCM}} = N + s^*,$$

where $N = \max(k, l) + 1$ is a constant independent of m, and s^* is the maximum number of satisfiable clauses in ϕ. By construction, maximizing OPT_{ADCM} is equivalent to maximizing s^*.

Contradiction via Approximation. Assume, for contradiction, that there exists a polynomial-time algorithm $\mathcal{A}$ that approximates ADCM within factor $\delta = \alpha + \varepsilon$. Let APX_{ADCM} be the solution returned by $\mathcal{A}$, so

$$APX_{\text{ADCM}} \geq \delta \cdot OPT_{\text{ADCM}}.$$

Consider the two gap cases:

- **YES-instance ($s^* = m$):**

$$APX_{\text{ADCM}} \geq (\alpha + \varepsilon)(N + m).$$

- **NO-instance ($s^* \leq (\alpha + \varepsilon/2)m$):**

$$APX_{\text{ADCM}} \leq OPT_{\text{ADCM}} \leq N + (\alpha + \varepsilon/2)m.$$

Separation Argument. The difference between the YES lower bound and the NO upper bound is

$$(\alpha + \varepsilon)(N + m) - \big(N + (\alpha + \varepsilon/2)m\big) = (\varepsilon - 1/8)N + \frac{\varepsilon}{2}m.$$

Since m can be arbitrarily large, this quantity is positive for all sufficiently large instances. Hence, by comparing APX_{ADCM} with the threshold

$$T = N + (\alpha + \varepsilon/2)m,$$

we can distinguish YES-instances from NO-instances in polynomial time. This contradicts the NP-hardness of the gap Max-3SAT problem.

Therefore, no polynomial-time algorithm can approximate ADCM within a factor better than $\boldsymbol{\alpha = 7/8}$, unless P = NP. □

EXPERIMENTAL RESULTS

Experimental results report based on real-world datasets .

We have conducted experiments on all datasets. Specifically, Figures 2 and 3 show the running times of all optimization algorithms on six real-world datasets, excluding the TC and WT datasets, for different values of k and (k, l); Figures 4 and 5 show the number of followers for all anchor selection strategies at different values of k and (k, l); Experimental results show that our candidate node pruning, score-based upper bound termination strategy, and our reuse technique can accelerate anchor acquisition time by at least three orders of magnitude compared to the greedy algorithm, with each optimization step generally improving the time by one order of magnitude. In addition, our designed score-based anchor selection method yields more followers than commonly used random point selection, degree-based selection and its variants, and the point selection strategy proposed by OLAK[7].

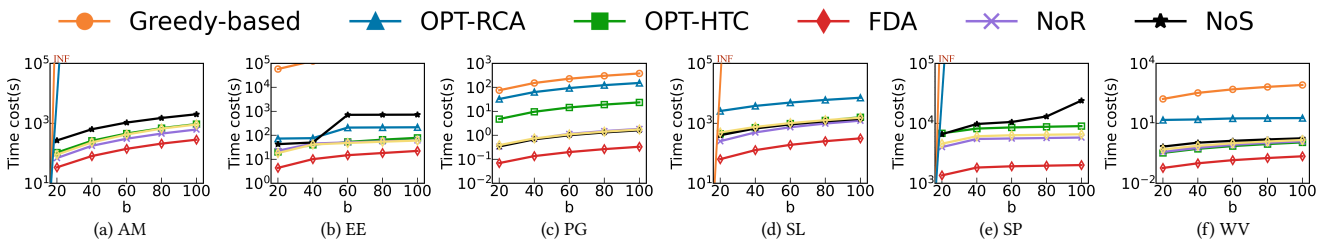

Fig. 1. Time cost under varying b

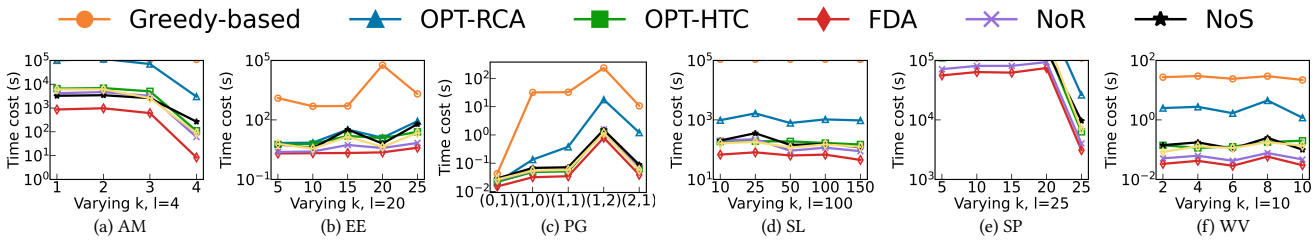

Fig. 2. Time cost under varying k

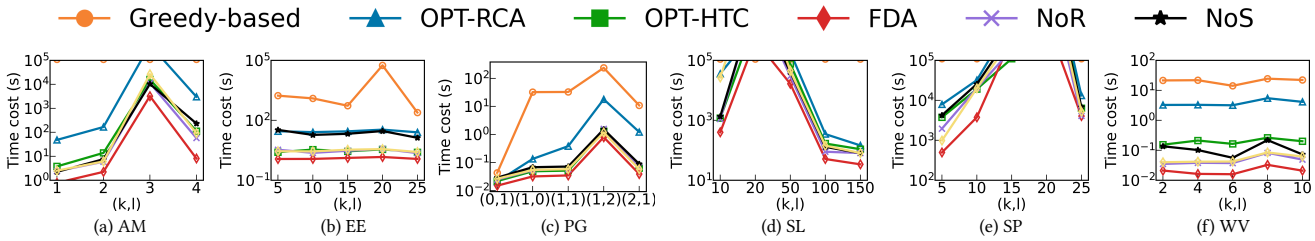

Fig. 3. Time cost under varying $(k, l), k = l$

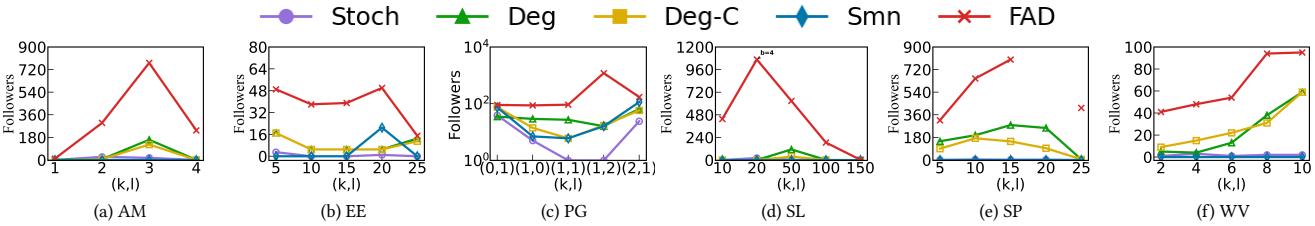

Fig. 4. Follower counts under varying $(k, l), k = l$

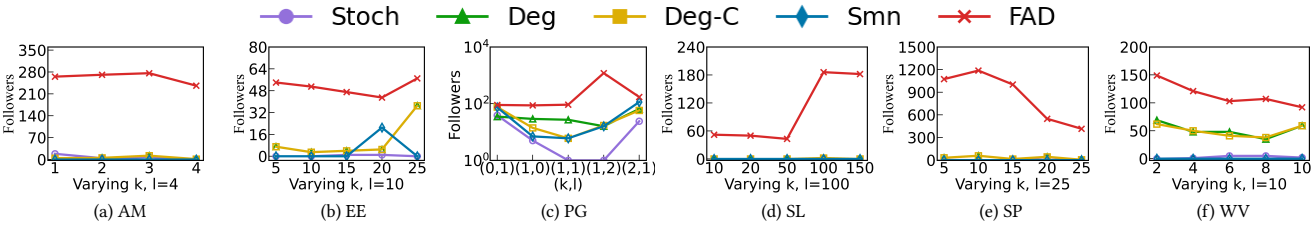

Fig. 5. Follower counts under varying $(k, l), k = l$

(k, l)-*Core Structure and Node Count Analysis for All Datasets* .

Since the number of datasets is quite large, and some of them contain cores exceeding the (200, 200)-core, the results cannot be clearly displayed within the limited page size. Therefore, we present the information in a zoomable web page available at: github.com.https://github.com/goodhy/FAD/blob/main/R1W3/ALL.html

Memory usage report .

Table 1 reports the peak memory consumption of the following components: Reuse: cached upper bounds and follower sets reused from the previous iteration. DSU (Disjoint Set Union): parent/rank arrays used to maintain connectivity within the $C_{k-1,l-1}/C_{k,l}$. Shell: hierarchical deletion order and layer information. Other: graph storage and temporary workspace.

Across datasets, the total memory footprint scales smoothly with graph size. DSU and OR together account for roughly 20% of the peak usage, as both structures are lightweight and stored only over shell vertices. The reuse mechanism prevents redundant reconstruction of

Table 1. Memory Footprint Breakdown Across All Datasets.

Component	AM (MB, %)	EE (MB, %)	PG (MB, %)	SL (MB, %)	SP (MB, %)	TC (MB, %)	WV (MB, %)	WT (MB, %)
Reuse	23.0 (21.8%)	10.0 (14.3%)	1.1 (26.8%)	171.5 (8.9%)	65.3 (8.8%)	5.5 (12.5%)	0.4 (12.9%)	87.1 (13.7%)
DSU	9.6 (9.1%)	9.1 (13.0%)	0.3 (7.3%)	166.4 (8.7%)	56.1 (7.6%)	2.8 (6.3%)	0.2 (6.5%)	82.2 (12.9%)
Shell	3.2 (3.0%)	2.8 (4.0%)	0.1 (2.4%)	50.9 (2.7%)	17.1 (1.0%)	0.9 (2.0%)	0.1 (3.2%)	25.1 (3.9%)
OR	9.4 (8.9%)	0.5 (0.7%)	0.5 (12.2%)	0.4 (0.02%)	7.1 (2.3%)	2.2 (5.0%)	0.1 (3.2%)	1.8 (0.3%)
Other	60.1 (57.0%)	47.5 (68.0%)	2.0 (51.2%)	1527.6 (79.6%)	596.1 (80.3%)	32.7 (74.1%)	2.3 (74.2%)	441.3 (69.2%)
Total	105.3	69.9	4.0	1916.8	741.7	44.1	3.1	637.5

temporary data across iterations, while shell decomposition restricts upper-bound propagation and follower exploration to a small subset of vertices.

Overall, the additional memory introduced by DSU and OR is moderate and well controlled, and our empirical analysis shows that this overhead remains small relative to the overall runtime while enabling the pruning and propagation mechanisms required by FAD.

All experimental results under all synthetic dataset .

We generated six synthetic datasets with varying graph sizes and densities to demonstrate the impact of data characteristics on algorithm performance through experimental evaluation.

Methods of synthesizing datasets. To examine the effect of dataset characteristics in a principled way, we first reviewed the scale and density statistics reported in prior studies [1, 5, 6] on real-world network datasets. Table 3 summarizes the typical parameter ranges observed in those datasets, including the number of vertices, edges, and average degree. Based on these empirical ranges, we generated six synthetic directed graphs using the power-law clustering (PLC) model implemented in **`NetworkX`** [2], covering different combinations of graph size and density. Furthermore, considering the hierarchical and dense nature of D-core structures, we introduced an onion-like layered architecture for generating each dataset. This architecture features: (1) dense inner layers and sparse outer layers - connections become increasingly dense toward the center; (2) inter-layer connectivity - each layer maintains limited connections with its inner adjacent layers; and (3) bidirectional flow - connections are bidirectional (implemented as directed edges in both directions). This design is essential to ensure that the datasets contain substantial (k, l)-core structures at higher k or l values, which would not be guaranteed with random or unidirectional connection patterns.

The specific configurations of these six datasets are listed in Table 2. While the number of vertices and average degree d_avg are fixed parameters, the number of edges varies depending on the graph generation process due to inherent imprecision. The $k_{\max}$ parameter represents the maximum out-degree, while $l_{\max}$ represents the maximum in-degree of core structures.

Table 2. Configuration settings of the six synthetic directed graphs

Dataset	Vertices	Edges	d_{avg}	$k_{\max}$	$l_{\max}$
PLC-S1	10,000	96,067	10	23	23
PLC-S2	10,000	194,034	20	30	30
PLC-M1	100,000	995,164	10	25	25
PLC-M2	100,000	1,992,508	20	30	30
PLC-L1	1,000,000	9,993,766	10	25	25
PLC-L2	1,000,000	19,990,117	20	30	30

Table 3. Typical parameter ranges of real-world networks

Type	Vertices	Edges	d_{avg}
Small	$< 10^4$	$< 10^5$	$3 \sim 20$
Medium	$10^4 \sim 10^6$	$10^5 \sim 10^7$	$1 \sim 25$
Large	$10^6 \sim 10^8$	$10^6 \sim 10^9$	$3 \sim 50$

Experimental conclusions. Tables 4 and 5 summarize the number of followers under different anchor selection strategies and the running times of various algorithms under default parameters, respectively. Tables 6 - 41 present the results of followers and running times when varying parameters b, k, l, and (k, l). Experimental results demonstrate that our score-based anchor selection strategy consistently achieves optimal number of follower. This finding aligns with the analysis on real-world datasets, confirming the effectiveness of our proposed algorithm.

In the ablation study conducted on synthetic datasets, although the FAD algorithm does not always deliver optimal results as it does on real-world datasets, it still outperforms the greedy algorithm in speed by 6 to 100 times. The reduced efficiency can be attributed to our graph generation approach, where only limited edges connect vertices across different hierarchical layers. This sparse inter-layer connectivity

leads to substantial time consumption in constructing data structures such as DSU and sequential reachable paths, along with repeated computational overhead. Consequently, the impact of cascade operations becomes less significant in such scenarios.

Note: In Tables 4 - 41, we consider running times exceeding 2 hours as timeout cases and mark them as INF.

Table 4. Running times under b=100

Alg.	Dataset					
	PLC-S1	PLC-S2	PLC-M1	PLC-M2	PLC-L1	PLC-L2
greedy	2.20353	1.12253	6502.84	1185.08	INF	INF
opt-rca	0.330542	0.232779	574.412	11.5426	34328	569.771
opt-htc	0.291167	0.236321	795.925	11.5631	35661.4	577.288
fad	0.35942	0.295777	901.81	15.7984	36070.3	767.339
NoR	0.316344	0.27733	903.996	12.7018	44662.6	756.83
NoC	0.69328	0.517547	1909.74	21.9113	INF	1378.04
NoU	0.410575	0.32624	680.593	11.7532	46127	756.741

Table 5. Follower count under b=100

Alg.	Dataset					
	PLC-S1	PLC-S2	PLC-M1	PLC-M2	PLC-L1	PLC-L2
Stoch	4	8	5	6	4	4
Deg	33	21	11	23	12	41
Deg-C	27	21	8	22	12	41
Smn	66	53	190	183	205	371
Loc-Opt	154	46	615	361	490	920
Our	161	92	593	285	490	1073

Table 6. Runtimes with varying k on $PLC-S1$

Alg.	k				
	5	10	15	20	25
greedy	28.2994	27.4366	33.3562	3.56715	0.042531
opt-rca	4.78132	3.24805	3.62118	0.271981	0.276087
opt-htc	2.43095	1.75449	2.20476	0.092466	0.070496
fad	2.3044	1.75152	1.20599	0.078177	0.065234
NoR	2.26706	1.62485	2.22644	0.07201	0.052829
NoC	5.45763	3.91078	9.54463	0.30319	0.190753
NoU	2.03886	1.84396	1.88935	0.100836	0.071648

Table 7. Runtimes with varying k on $PLC-S2$

Alg.	k				
	5	10	15	20	25
greedy	96.7608	93.8873	89.7147	11.3821	10.5235
opt-rca	20.6849	18.9055	17.9209	0.174865	0.181379
opt-htc	14.335	14.2247	11.3771	0.08744	0.095797
fad	11.3274	11.193	11.5723	0.100307	0.109026
NoR	13.6523	13.716	10.915	0.066916	0.074579
NoC	20.0091	19.914	18.1343	0.111551	0.250131
NoU	12.9264	11.8405	11.196	0.092528	0.100094

Table 8. Runtimes with varying k on $PLC-M1$

Alg.	k				
	5	10	15	20	25
greedy	INF	INF	4637.75	6283.61	76.7464
opt-rca	751.791	735.701	152.249	99.232	2.86747
opt-htc	531.883	539.099	153.615	63.3196	1.93513
fad	529.721	538.11	124.631	64.7478	1.93528
NoR	407.484	400.914	117.145	49.6437	1.31054
NoC	1024.5	992.613	557.891	871.167	2.58016
NoU	455.82	459.085	121.283	55.7889	1.51871

Table 9. Runtimes with varying k on $PLC-M2$

Alg.	k				
	5	10	15	20	25
greedy	INF	INF	INF	5635.38	4864.06
opt-rca	3437.12	3352.98	1168.58	4.81319	3.41309
opt-htc	2558.57	2488.86	711.395	4.47569	3.23084
fad	2592.81	2593.1	609.361	4.06765	3.18238
NoR	2531.85	2549.18	671.136	5.27131	3.80886
NoC	3609.02	3579.42	1993.7	7.55542	4.30492
NoU	2393.36	2391.65	657.321	5.13989	3.20274

Table 10. Runtimes with varying k on $PLC-L1$

Alg.	k				
	5	10	15	20	25
greedy	INF	INF	INF	INF	1955.39
opt-rca	INF	INF	INF	INF	29.65
opt-htc	INF	INF	INF	INF	23.5607
fad	INF	INF	INF	INF	19.6785
NoR	INF	INF	INF	INF	17.4756
NoC	INF	INF	INF	INF	23.6699
NoU	INF	INF	INF	INF	19.9478

Table 11. Runtimes with varying k on $PLC-L1$

Alg.	k				
	5	10	15	20	25
greedy	INF	INF	INF	INF	INF
opt-rca	INF	INF	INF	127.34	159.119
opt-htc	INF	INF	INF	111.669	126.523
fad	INF	INF	INF	110.145	151.462
NoR	INF	INF	INF	117.486	165.208
NoC	INF	INF	INF	174.23	202.003
NoU	INF	INF	INF	102.579	124.38

Table 12. Runtimes with varying l on $PLC - S1$

Alg.	l				
	5	**10**	**15**	**20**	**25**
greedy	23.67	23.558	3.4752	3.54447	0.040745
opt-rca	6.15603	5.56843	1.851	0.200074	0.186113
opt-htc	2.92738	2.92031	0.279781	0.091132	0.070215
fad	2.80365	2.6907	0.26674	0.068428	0.068014
NoR	2.7439	2.68664	0.270202	0.068461	0.053262
NoC	12.3434	8.53306	0.693231	0.267301	0.166072
NoU	3.87913	3.89552	1.07656	0.105596	0.091593

Table 13. Runtimes with varying l on $PLC - S2$

Alg.	l				
	5	**10**	**15**	**20**	**25**
greedy	95.6794	114.868	55.6568	10.9125	12.7748
opt-rca	25.3603	25.3587	4.14494	0.177145	0.229656
opt-htc	17.4156	17.6163	2.26001	0.089168	0.131616
fad	18.0837	17.5775	2.41139	0.099341	0.104142
NoR	16.9684	17.3175	2.17552	0.071604	0.109057
NoC	32.5051	25.0435	10.4587	0.118645	0.12424
NoU	16.6524	16.4756	2.60753	0.09085	0.132685

Table 14. Runtimes with varying l on $PLC - M1$

Alg.	l				
	5	**10**	**15**	**20**	**25**
greedy	INF	INF	INF	6606.57	53.7758
opt-rca	1182.27	1153.47	559.835	67.5569	1.69193
opt-htc	966.105	927.037	411.768	59.7071	1.74266
fad	811.229	880.9	380.632	63.8762	1.68364
NoR	746.583	697.168	339.432	50.5029	1.34527
NoC	1613.4	1626.3	1253.34	827.903	1.71791
NoU	836.073	809.402	386.284	57.4881	1.6079

Table 15. Runtimes with varying l on $PLC - M2$

Alg.	l				
	5	**10**	**15**	**20**	**25**
greedy	INF	INF	INF	6501.44	3812.98
opt-rca	3095.68	3009.37	1173.49	4.69993	3.02377
opt-htc	2499.86	2491.49	1008.81	4.3723	2.71732
fad	2795.08	2992.17	1453.16	5.43321	3.72074
NoR	2667.3	2806.09	1125.65	4.84499	2.82733
NoC	3289.64	3328.57	1768.25	7.54922	4.2995
NoU	2414.08	2447.55	1006.33	4.71619	2.85054

Table 16. Runtimes with varying l on $PLC - L1$

Alg.	l				
	5	**10**	**15**	**20**	**25**
greedy	INF	INF	INF	INF	1761.38
opt-rca	INF	INF	INF	INF	29.8659
opt-htc	INF	INF	INF	INF	27.0775
fad	INF	INF	INF	INF	20.0501
NoR	INF	INF	INF	INF	21.3504
NoC	INF	INF	INF	INF	22.8893
NoU	INF	INF	INF	INF	20.4359

Table 17. Runtimes with varying l on $PLC - L2$

Alg.	l				
	5	**10**	**15**	**20**	**25**
greedy	INF	INF	INF	INF	INF
opt-rca	INF	INF	INF	125.876	135.01
opt-htc	INF	INF	INF	106.935	127.282
fad	INF	INF	INF	139.903	147.125
NoR	INF	INF	INF	129.438	143.451
NoC	INF	INF	INF	207.142	238.89
NoU	INF	INF	INF	107.166	111.562

Table 18. Runtimes with varying (k, l) on PLC-S1

Alg.	(k, l), k=l				
	5	**10**	**15**	**20**	**25**
greedy	104.033	24.8695	3.82114	3.60084	0.052618
opt-rca	70.9177	3.14577	0.170536	0.172431	0.143073
opt-htc	43.315	2.24953	0.083932	0.091235	0.070342
fad	29.7729	1.60272	0.074437	0.07669	0.087734
NoR	42.3195	2.14532	0.06205	0.06922	0.062996
NoC	116.552	5.30727	0.107031	0.151119	0.085261
NoU	46.1457	1.98769	0.092512	0.105261	0.088839

Table 19. Runtimes with varying (k, l) on PLC-S2

Alg.	(k, l), k=l				
	5	**10**	**15**	**20**	**25**
greedy	47.1685	56.1889	3.8275	11.7664	102.045
opt-rca	18.5762	6.88989	1.00756	0.170508	10.5633
opt-htc	9.58324	3.61583	0.716624	0.093958	7.97729
fad	5.84418	3.06716	0.233023	0.100994	7.77636
NoR	9.49524	3.62269	0.591454	0.068766	7.36723
NoC	25.895	6.26575	3.71117	0.211504	12.9566
NoU	10.9388	3.83811	0.569988	0.089941	6.8693

Table 20. Runtimes with varying (k, l) on PLC-M1

Alg.	(k, l), k=l				
	5	**10**	**15**	**20**	**25**
greedy	INF	6879.59	3849.73	INF	246.316
opt-rca	INF	50.8693	10.4905	73.7517	2.32965
opt-htc	INF	45.2093	9.84305	59.7265	1.71564
fad	INF	27.9633	8.12728	60.4389	1.64684
NoR	INF	37.7057	8.1206	53.473	1.39607
NoC	INF	387.014	15.6879	994.382	2.50991
NoU	INF	43.9992	9.15211	56.9714	1.63576

Table 21. Runtimes with varying (k, l) on PLC-M2

Alg.	(k, l), k=l				
	5	**10**	**15**	**20**	**25**
greedy	INF	INF	INF	5265.52	INF
opt-rca	4557.21	3533.26	15.6113	4.6531	INF
opt-htc	3810.64	2412.89	14.0847	5.12402	INF
fad	2940.16	2640.47	17.8878	5.8391	INF
NoR	5305.7	3510.68	15.6147	4.72576	INF
NoC	INF	3123.75	141.147	7.45015	INF
NoU	4271.57	2756.11	13.2284	4.41038	INF

Table 22. Runtimes with varying (k, l) on PLC-L1

Alg.	(k, l), k=l				
	5	10	15	20	25
greedy	INF	INF	INF	INF	5883.85
opt-rca	INF	INF	2062.78	INF	29.6222
opt-htc	INF	INF	1793.94	INF	22.7625
fad	INF	INF	1272.6	INF	22.4357
NoR	INF	INF	1702.5	INF	18.1351
NoC	INF	INF	3512.8	INF	21.5316
NoU	INF	INF	1655.91	INF	20.7293

Table 23. Runtimes with varying (k, l) on PLC-L2

Alg.	(k, l), k=l				
	5	10	15	20	25
greedy	INF	INF	INF	INF	INF
opt-rca	INF	INF	426.139	138.782	INF
opt-htc	INF	INF	368.642	106.073	INF
fad	INF	INF	212.93	115.696	INF
NoR	INF	INF	612.585	156.218	INF
NoC	INF	INF	INF	244.395	INF
NoU	INF	INF	365.448	110.637	INF

Table 24. Follower count under b=20 varying k, l=20,dataset=PLC-S1

Alg.	k				
	5	10	15	20	25
deg	11	0	52	7	0
degC	7	0	52	3	0
stoch	16	0	0	12	0
Smn	38	1	69	28	INF
fad	165	36	82	29	0

Table 25. Follower count under b=20 varying k, l=20,dataset=PLC-S2

Alg.	k				
	5	10	15	20	25
deg	1	1	1	8	5
degC	1	1	1	8	5
stoch	0	0	2	1	0
Smn	15	15	4	24	21
fad	156	156	19	40	36

Table 26. Follower count under b=20 varying k, l=20,dataset=PLC-M1

Alg.	k				
	5	10	15	20	25
deg	2	2	1	0	0
degC	2	2	1	0	0
stoch	7	0	0	0	0
Smn	31	31	9	12	17
fad	218	148	176	75	23

Table 27. Follower count under b=20 varying k, l=20,dataset=PLC-M2

Alg.	k				
	5	10	15	20	25
deg	0	0	0	2	2
degC	0	0	0	2	2
stoch	0	0	0	0	0
Smn	17	16	6	68	64
fad	56	56	59	130	126

Table 28. Follower count under b=20 varying k, l=20,dataset=PLC-L1

Alg.	k				
	5	10	15	20	25
deg	2	2	5	1	0
degC	2	2	5	1	0
stoch	0	0	1	0	0
Smn	42	36	7	34	12
fad	INF	INF	INF	INF	21

Table 29. Follower count under b=20 varying k, l=20,dataset=PLC-L2

Alg.	k				
	5	10	15	20	25
deg	6	5	5	5	1
degC	6	5	5	5	1
stoch	0	0	2	0	0
Smn	32	22	6	102	89
fad	INF	INF	INF	297	293

Table 30. Follower count with varying l on PLC-S1

Alg.	l				
	5	10	15	20	25
deg	26	26	98	7	0
degC	26	26	98	3	0
stoch	5	0	1	12	0
Smn	26	26	67	28	INF
fad	159	155	109	29	0

Table 31. Follower count with varying l on PLC-S2

Alg.	l				
	5	10	15	20	25
deg	2	2	2	8	0
degC	2	2	2	8	0
stoch	2	1	0	0	2
Smn	15	15	10	24	23
fad	20	20	24	40	26

Table 32. Follower count with varying l on PLC-M1

Alg.	l				
	5	10	15	20	25
deg	1	1	1	0	1
degC	1	1	1	0	1
stoch	0	0	0	0	0
Smn	33	28	21	12	16
fad	281	85	77	75	18

Table 33. Follower count with varying l on PLC-M2

Alg.	l				
	5	10	15	20	25
deg	5	4	4	2	2
degC	5	4	4	2	2
stoch	1	4	1	0	0
Smn	71	52	19	68	39
fad	69	68	67	130	119

Table 34. Follower count with varying l on PLC-L1

Alg.	l				
	5	10	15	20	25
deg	14	14	14	1	0
degC	14	14	14	1	0
stoch	1	0	6	0	0
Smn	91	55	29	34	12
fad	INF	INF	INF	INF	19

Table 35. Follower count with varying l on PLC-L2

Alg.	l				
	5	10	15	20	25
deg	1	0	0	5	3
degC	1	0	0	5	3
stoch	2	0	0	0	0
Smn	78	61	23	102	82
fad	INF	INF	INF	297	288

Table 36. Follower count with varying (k, l) on PLC-S1

Alg.	(k, l), k=l				
	5	10	15	20	25
deg	11	0	52	7	0
degC	7	0	52	3	0
stoch	16	0	0	12	0
Smn	38	1	69	28	INF
fad	165	36	82	29	0

Table 37. Follower count with varying (k, l) on PLC-S2

Alg.	(k, l), k=l				
	5	10	15	20	25
deg	33	2	0	8	10
degC	35	2	0	8	10
stoch	21	1	1	3	1
Smn	41	107	11	24	25
fad	118	199	37	40	296

Table 38. Follower count with varying (k, l) on PLC-M1

Alg.	(k, l), k=l				
	5	10	15	20	25
deg	64	6	2	0	1
degC	72	6	2	0	1
stoch	37	0	1	2	0
Smn	77	15	144	12	28
fad	INF	76	298	75	42

Table 39. Follower count with varying (k, l) on PLC-M2

Alg.	(k, l), k=l				
	5	10	15	20	25
deg	103	0	1	2	37
degC	103	0	1	2	37
stoch	1	2	1	0	0
Smn	111	241	17	68	136
fad	306	1354	71	130	INF

Table 40. Follower count with varying (k, l) on PLC-L1

Alg.	(k, l), k=l				
	5	10	15	20	25
deg	83	1	2	1	0
degC	93	1	2	1	0
stoch	1	0	0	0	0
Smn	117	24	138	34	21
fad	INF	INF	462	INF	36

Table 41. Follower count with varying (k, l) on PLC-L2

Alg.	(k, l), k=l				
	5	10	15	20	25
deg	91	1	1	5	5
degC	85	1	1	5	5
stoch	2	1	0	0	3
Smn	147	748	3	102	26
fad	INF	INF	108	297	INF

References

[1] Yixiang Fang, Reynold Cheng, Yankai Chen, Siqiang Luo, and Jiafeng Hu. 2017. Effective and efficient attributed community search. VLDB J. 26, 6 (2017), 803–828. https://doi.org/10.1007/S00778-017-0482-5

[2] Aric Hagberg, Pieter J Swart, and Daniel A Schult. 2008. Exploring network structure, dynamics, and function using NetworkX. Technical Report. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).

[3] Johan Håstad. 1997. Some Optimal Inapproximability Results. In Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997, Frank Thomson Leighton and Peter W. Shor (Eds.). ACM, 1–10. https://doi.org/10.1145/258533.258536

[4] H. Karloff and U. Zwick. 1997. A 7/8-approximation algorithm for MAX 3SAT?. In Proceedings 38th Annual Symposium on Foundations of Computer Science. 406–415. https://doi.org/10.1109/SFCS.1997.646129

[5] Xuankun Liao, Qing Liu, Jiaxin Jiang, Xin Huang, Jianliang Xu, and Byron Choi. 2022. Distributed D-core Decomposition over Large Directed Graphs. Proc. VLDB Endow. 15, 8 (2022), 1546–1558. https://doi.org/10.14778/3529337.3529340

[6] Wensheng Luo, Yixiang Fang, Chunxu Lin, and Yingli Zhou. 2024. Efficient Parallel D-Core Decomposition at Scale. Proc. VLDB Endow. 17, 10 (June 2024), 2654–2667. https://doi.org/10.14778/3675034.3675054

[7] Fan Zhang, Wenjie Zhang, Ying Zhang, Lu Qin, and Xuemin Lin. 2017. OLAK: an efficient algorithm to prevent unraveling in social networks. Proc. VLDB Endow. 10, 6 (feb 2017), 649–660. https://doi.org/10.14778/3055330.3055332