Residual Core Maximization: An Efficient Algorithm for Maximizing the Size of the k-Core (Supplementary Material)

Ricky Laishram* Ahmet Erdem Sarıyüce† Tina Eliassi-Rad‡ Ali Pinar§
Sucheta Soundarajan*

1 Proofs

LEMMA 1.1. *Nodes that are not in C_f cannot become followers. That is, $\forall v \in \left(\overline{V_{k,A}} \setminus C_f\right), \nexists A' \subseteq \overline{V_{k,A}}$ such that $v \in \mathcal{F}(k, A \cup A')$.*

Proof. Since, $v \notin C_f$, $|N(v)| < k$. If $v \in \mathcal{F}(k, A \cup A')$, by definition $|N(v) \cap V_{k,A\cup A'}| \geq k \implies |N(v)| \geq k$. This is a contradiction. So, $v \notin \mathcal{F}(k, A \cup A')$. ∎

LEMMA 1.2. *Adding any subset of $\overline{V_{k,A}} \setminus C_a$ to the set of anchors will not change the set of followers. That is, $\forall A' \subseteq \left(\overline{V_{k,A}} \setminus C_a\right)$, $\mathcal{F}(k, A) = \mathcal{F}(k, A \cup A')$.*

Proof. Consider $A' \subseteq \left(\overline{V_{k,A}} \setminus C_a\right)$. It is easy to show that $\mathcal{F}(k, A) \subseteq \mathcal{F}(k, A \cup A')$. So,

$$\mathcal{F}(k, A) \setminus \mathcal{F}(k, A \cup A') = \emptyset. \tag{1.1}$$

Let $D = \mathcal{F}(k, A \cup A') \setminus \mathcal{F}(k, A)$. By Lemma 1.1, $D \subseteq C_f \subseteq \overline{V_{k,A}}$. Then by the definition of anchored k-core, $\forall v \in D$,

$$\begin{aligned} |N(v) \cap V_{k,A\cup A'}| &\geq k \\ |N(v) \cap (V_{k,A} \cup A' \cup D)| &\geq k \\ |N(v) \cap (V_{k,A} \cup D)| + |N(v) \cap A'| &\geq k. \end{aligned}$$

Because $A' \cap C_a = \emptyset$, $\nexists u \in A'$ such that $u \in N(v)$ (by definition of C_a). So, $|N(v) \cap A'| = 0$. Then,

$$|N(v) \cap (V_{k,A} \cup D)| \geq k.$$

This means that $V_{k,A} \cup D$ is the set of nodes in the anchored k-core with anchors A, because by definition, the anchored k-core is the maximal set. So all the nodes that are in the anchored k-core with anchors $A \cup A'$ are already in the set $V_{k,A}$. Then, $D = \emptyset$.

$$\mathcal{F}(k, A \cup A') \setminus \mathcal{F}(k, A) = D = \emptyset. \tag{1.2}$$

Therefore, from (1.1) and (1.2), we get $\mathcal{F}(k, A) = \mathcal{F}(k, A \cup A')$. ∎

LEMMA 1.3. *Residual anchor selection in NP-hard.*

Proof. We will show this by reducing the set cover problem to the residual anchor selection problem. Suppose we have a set cover problem with finite sets $U \subseteq \mathbb{Z}_+$ and $S = \{S_0, S_1, \ldots\}$ such that $S_i \subseteq U$.. The set cover problem is to find the set S^* such that,

$$\begin{aligned} S^* &= \arg\min_{S' \subseteq \overline{S}} |S'| \\ \text{s.t.} &\bigcup_{X \in S'} X = U \end{aligned}$$

Let us generate the following,

$$\begin{aligned} R &= \{0, 1, \ldots, |S| - 1\} \\ E &= \{(i, j) : i \in U \wedge i \in S_j\}. \end{aligned}$$

Now we can construct a bipartite graph $B = \langle U, R, E\rangle$. By construct, there is a one-to-one mapping between R and S. So, $(i, j) \in E$ denotes the membership of $i \in U$ to $S_j \in S$. So, with this this construction, the set cover problem can be stated as: find R^* such that,

$$\begin{aligned} R^* &= \arg\min_{R' \subseteq R} |R'| \\ \text{s.t.} &\bigcup_{r \in R^*} N(r, B) = U. \end{aligned}$$

If we have $\delta' : U \to 1$, the problem has reduced to the residual anchor selection, where U and R correspond to V_o' and $C_a \setminus C_f$. So the residual anchor selection problem in NP-hard. ∎

*Syracuse University, Syracuse, NY, USA. Email: (*rlaishra, susounda@syr.edu*)
†University at Buffalo, Buffalo, NY, USA. Email: (*erdem@buffalo.edu*)
‡Northeastern University, Boston, MA, USA. Email: (*eliassi@ccs.neu.edu*)
§Sandia National Laboratories, Livermore, CA, USA. Email: (*apinar@sandia.gov*)

2 Running Time of RCM

In this section we will discuss the running time of RCM. We begin by discussing the running time of the various components described so far.

Selecting Candidate Anchors: Selection of candidate anchors requires only counting the neighbors of nodes in $\overline{V_{k,A}}$. So, C_f and C_a can be found in $O\left(|\overline{V_{k,A}}|\right)$.

Residual Degree: To find the residual degree, we need to count neighbors of all the nodes in C_f. This can be done in $O\left(|C_f|\right)$.

Connected Components: The connected components of G_f can be found in $O\left(|E_f|\right)$, where E_f is the set of edges in G_f.

Bound on Number of Anchors: For a component $G' \in \mathcal{G}$, we first need to find the set of nodes V'_o and V'_i. This requires only counting the number of neighbors of the nodes in G'. So, it can be done in $O\left(|V'|\right)$. Then we need to count the neighbors of V'_o to find $\beta^{\top}\left(G'\right)$, $\beta^{\perp}\left(G'\right)$ and $\beta^{*}\left(G'\right)$. The running time of this step is $O\left(|V'_o|\right)$. Then, the overall running time for the component G' is $O\left(V'\right)$. Since we need to find the bounds for all the components, the total running time is $O\left(|C_f|\right)$.

Residual Anchors: In Algorithm 2 (main paper), we need to check for anchors in $\left(C_a \setminus C_f\right) \cap N\left(V'_o\right)$. The number of iterations in the algorithm is of the order of $|V'_o|$ and $|\left(C_a \setminus C_f\right) \cap N\left(V'_o\right)| \leq \beta^{\top}\left(G'\right)$. So, the running time for component G' is $O\left(\beta^{\top}\left(G'\right)|V'_o|\right)$. Assuming that we need to find the residual anchors for all the components, the running time is $O\left(\sum_{G' \in \mathcal{G}} \beta^{\top}\left(G'\right)|V'_o|\right) \approx O\left(|C_f|\right)$.

Anchor Score based Anchors: For a component G', to find the Anchor Score of all the nodes in $C'_f \cup C'_a$. This can be done in $O\left(|E'_{fa}|\right)$, where E'_{fa} is the set of anchors in the induced subgraph of $C'_f \cup C'_a$. We then need to find the followers of the selected anchor with `FindResidualCore()` and this takes $O\left(|C'_f|\right)$. Then, if we consider all the components, the time to find b anchors is $O\left(b \cdot \left(|E_{fa}| + |C_a|\right)\right) \approx O\left(b \cdot |E_{fa}|\right)$, where E_{fa} is the set of edges in the induced subgraph of $C_f \cup C_a$.

Overall Running Time: By combining the running time of all different parts, we can get the overall running time of RCM as,

$$O\left(|\overline{V_{k,A}}| + |C_f| + |E_f| + |C_f| + |E_{fa}|\right) \approx O\left(|E_{fa}|\right).$$

3 Results

Table 1 shows the complete results for all the networks we consider for fixed k (at the median value) and $k = 50, 100, 150, 200, 250$. We can see that in all the cases, RCM finds the most followers and is much faster at all values of b.

Figure 1 shows the followers at fixed b (at 100) and k corresponding to the 30^{th}, 45^{th}, 60^{th}, 75^{th}, and 90^{th} percentiles. Again, RCM outperforms all the baselines for all the k values.

Network	Alg.	Followers 50	100	150	200	250	Time (ms) 50	100	150	200	250
FC	RCM	180.0	217.0	293.0	338.0	412.0	41.6	28.0	1.1	1.0	0.8
	OLAK	39.0	89.0	100.0	115.0	165.0	1768.9	1539.7	2066.4	2417.8	2119.1
	MD	45.8	64.6	92.8	115.0	144.8	15187.0	13541.3	10105.5	8775.8	6355.4
	RND	37.0	119.0	155.0	191.0	222.0	21962.6	4220.9	3708.3	3213.5	2927.5
CC	RCM	137.0	237.0	320.0	390.0	447.0	35.7	21.1	15.5	13.0	11.2
	OLAK	95.0	151.0	213.0	267.0	299.0	1289.1	1604.3	1697.0	1800.9	2008.0
	MD	57.8	102.6	145.2	182.8	225.4	17168.7	10006.9	7200.3	6246.2	5080.3
	RND	24.0	35.0	62.0	95.0	122.0	92435.2	86366.6	41562.2	23676.4	17962.6
CH	RCM	130.0	215.0	275.0	326.0	375.0	10.5	6.5	5.1	4.3	3.8
	OLAK	103.0	155.0	195.0	234.0	259.0	1075.8	1361.9	1586.3	1741.1	1957.6
	MD	61.2	99.0	139.8	186.0	211.0	11740.0	9002.4	6471.8	5017.8	4877.7
	RND	22.0	42.0	75.0	92.0	115.0	94470.9	51864.9	24389.9	21585.2	17245.7
LB	RCM	160.0	260.0	360.0	460.0	560.0	94.5	58.5	42.6	33.7	27.9
	OLAK	123.0	216.0	299.0	375.0	449.0	2395.0	2519.9	2643.3	2771.2	2861.3
	MD	47.4	92.4	138.4	185.0	232.6	50192.5	26053.7	17049.6	12768.4	10303.2
	RND	43.0	85.0	130.0	179.0	227.0	64900.4	33244.6	21312.7	14970.3	11626.0
FN	RCM	100.0	160.0	225.0	285.0	342.0	729.8	792.0	774.3	757.1	752.9
	OLAK	47.0	70.0	104.0	148.0	167.0	6489.6	8392.8	8311.0	7719.9	8534.2
	MD	56.2	97.2	140.6	188.4	221.8	47620.6	31166.7	21183.7	16512.8	15078.4
	RND	22.0	47.0	67.0	86.0	97.0	285906.3	125623.1	92414.7	74769.9	73501.1
FS	RCM	102.0	169.0	264.0	313.0	372.0	1230.1	1314.1	1173.0	1209.8	1188.7
	OLAK	52.0	65.0	101.0	136.0	163.0	8138.1	12833.8	11923.2	11692.3	12167.6
	MD	52.6	84.2	131.2	179.2	218.0	69381.9	56295.1	33511.5	24215.2	20737.2
	RND	13.0	39.0	51.0	62.0	81.0	1190057.8	263336.2	229540.8	205984.3	150431.7
CS	RCM	193.0	343.0	489.0	612.0	711.0	1448.1	829.9	571.7	465.6	393.8
	OLAK	150.0	248.0	331.0	429.0	514.0	9154.2	10684.0	11777.9	11983.6	12384.4
	MD	24.4	59.6	91.8	134.8	175.8	1042825.4	366797.9	229986.8	136228.6	105436.8
	RND	26.0	51.0	86.0	107.0	128.0	901144.1	468652.4	248106.3	214736.7	187815.6
LG	RCM	221.0	390.0	542.0	692.0	842.0	1520.9	859.6	622.4	490.4	402.0
	OLAK	163.0	290.0	419.0	532.0	640.0	7996.8	8572.8	8665.3	8994.1	9264.6
	MD	57.4	113.0	168.6	226.2	284.6	172273.5	91977.7	60466.1	45490.1	35338.2
	RND	47.0	86.0	133.0	172.0	210.0	259903.1	155362.4	97168.8	77412.3	64894.3
KD	RCM	197.0	346.0	497.0	636.0	746.0	2450.3	1374.6	967.8	766.3	658.8
	OLAK	113.0	231.0	320.0	403.0	501.0	15805.8	14787.5	15759.4	16540.3	16555.4
	MD	41.8	75.8	114.0	164.2	213.6	503756.6	316220.8	200452.2	120634.6	89436.6
	RND	25.0	62.0	95.0	119.0	160.0	1354643.6	440268.7	281311.1	239019.7	165546.5
SC	RCM	166.0	256.0	328.0	394.0	470.0	122.1	77.9	61.4	51.0	43.4
	OLAK	66.0	132.0	195.0	214.0	253.0	70318.4	68169.6	68413.3	82560.0	87389.5
	MD	98.8	161.4	217.8	282.0	353.4	175485.6	132589.8	112044.5	87978.3	68826.0
	RND	15.0	61.0	89.0	109.0	122.0	7875766.3	936644.4	649948.8	573095.7	568578.5
SD	RCM	207.0	367.0	502.0	607.0	716.0	1756.8	988.9	720.6	595.9	508.1
	OLAK	74.0	142.0	193.0	265.0	308.0	120296.1	124644.4	137108.9	132946.6	142881.3
	MD	88.0	172.0	257.8	315.2	360.8	458973.7	246258.1	164010.3	137813.0	126279.2
	RND	17.0	40.0	66.0	87.0	121.0	12001747.6	4179557.2	2259650.0	1713637.4	1096445.9
ST	RCM	210.0	313.0	417.0	519.0	621.0	2943.0	2116.5	1835.9	1446.7	1350.9
	OLAK	51.0	67.0	104.0	123.0	150.0	274560.0	417463.9	406547.5	456401.2	462212.4
	MD	37.6	97.8	156.6	198.2	246.8	3107387.4	987846.1	555503.8	490668.2	378017.0
	RND	38.0	57.0	77.0	101.0	127.0	3239471.6	2879554.6	2372091.7	1838582.1	1452889.3
WH	RCM	279.0	476.0	678.0	852.0	1008.0	27406.3	15907.3	11254.1	8917.5	7523.1
	OLAK	125.0	203.0	296.0	370.0	491.0	157331.7	193736.3	200184.4	210121.9	194232.0
	MD	61.6	124.8	202.5	267.3	340.9	1610198.8	816209.0	460616.9	328244.7	248445.7
	RND	32.2	63.6	97.4	130.1	158.4	15735043.8	7206615.4	4948714.6	3711509.7	3090567.6
WB	RCM	235.0	427.0	577.0	734.0	882.0	16575.5	9054.2	6730.3	5246.3	4432.9
	OLAK	148.0	286.0	378.0	501.0	585.0	150912.9	150811.0	169821.7	170291.5	181433.2
	MD	50.4	113.0	180.0	239.6	300.4	4102807.3	1426042.4	811311.5	597483.2	472545.4
	RND	37.0	80.0	110.0	139.0	170.0	6837427.8	2903604.5	2294884.6	1911577.3	1590125.3

Table 1: Comparison between RCM and various baseline algorithms for fixed k (median value). The number of followers found and the time to find each follower is indicated under the the budget used. The different columns gives the number of follower and time efficiency for $b = 50, 100, 150, 200, 250$. For followers count higher values are better and for the time lower values are better.

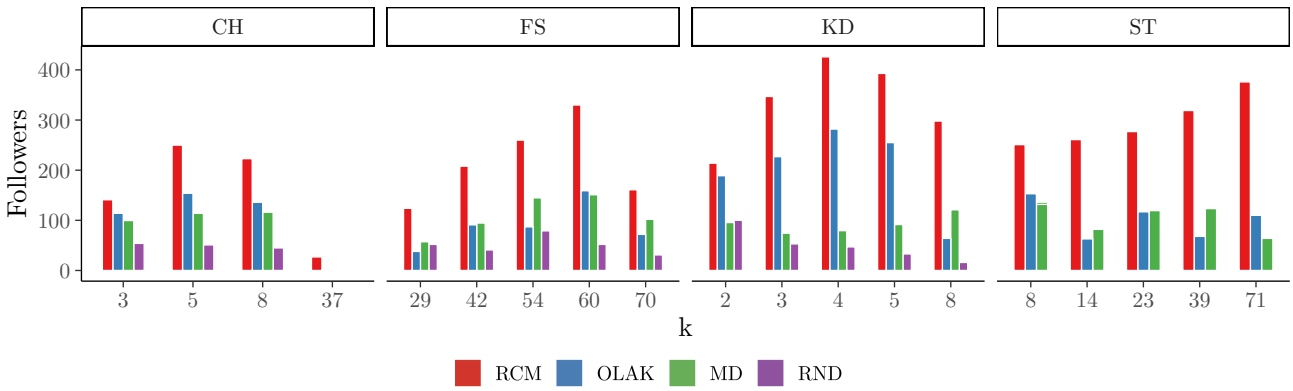

Figure 1: Number of followers found by `RCM` and various baseline algorithms for different values of k (and $b = 100$). We can observe that `RCM` performs the best for all values of k considered. In the case of `ST`, there are only 3 bars because `RND` does not find any followers.