Three-step Algorithms for Detection of High Degree Nodes in Online Social Networks

Danil Shaikhelislamov[1], Mikhail Drobyshevskiy[2], Denis Turdakov[2,3], Alexander Yatskov[2], Maksim Varlamov[2], Denis Aivazov[1,2]

[1]Moscow Institute of Physics and Technology (State University), Moscow, Russia
[2]Ivannikov Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia
[3]Lomonosov Moscow State University, Moscow, Russia

IVANNIKOV MEMORIAL WORKSHOP, OREL
SEPTEMBER 25-26, 2020

Plan

ISP RAS

1. Introduction
2. Description of 3-Step and 3-StepBatch
3. Optimal parameters
4. Experiments
5. Conclusion

What? Online Social Network

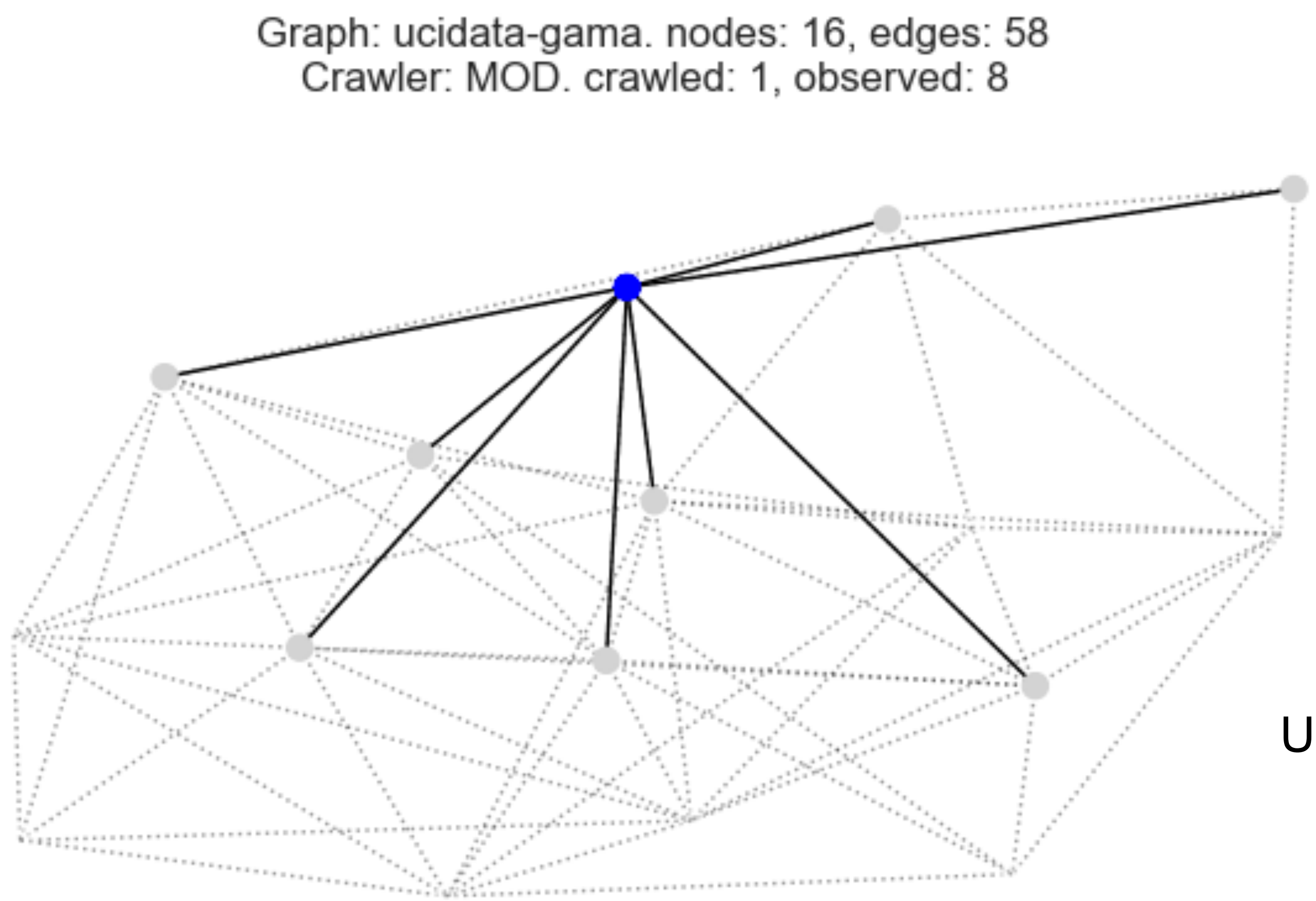
Nodes represent the users

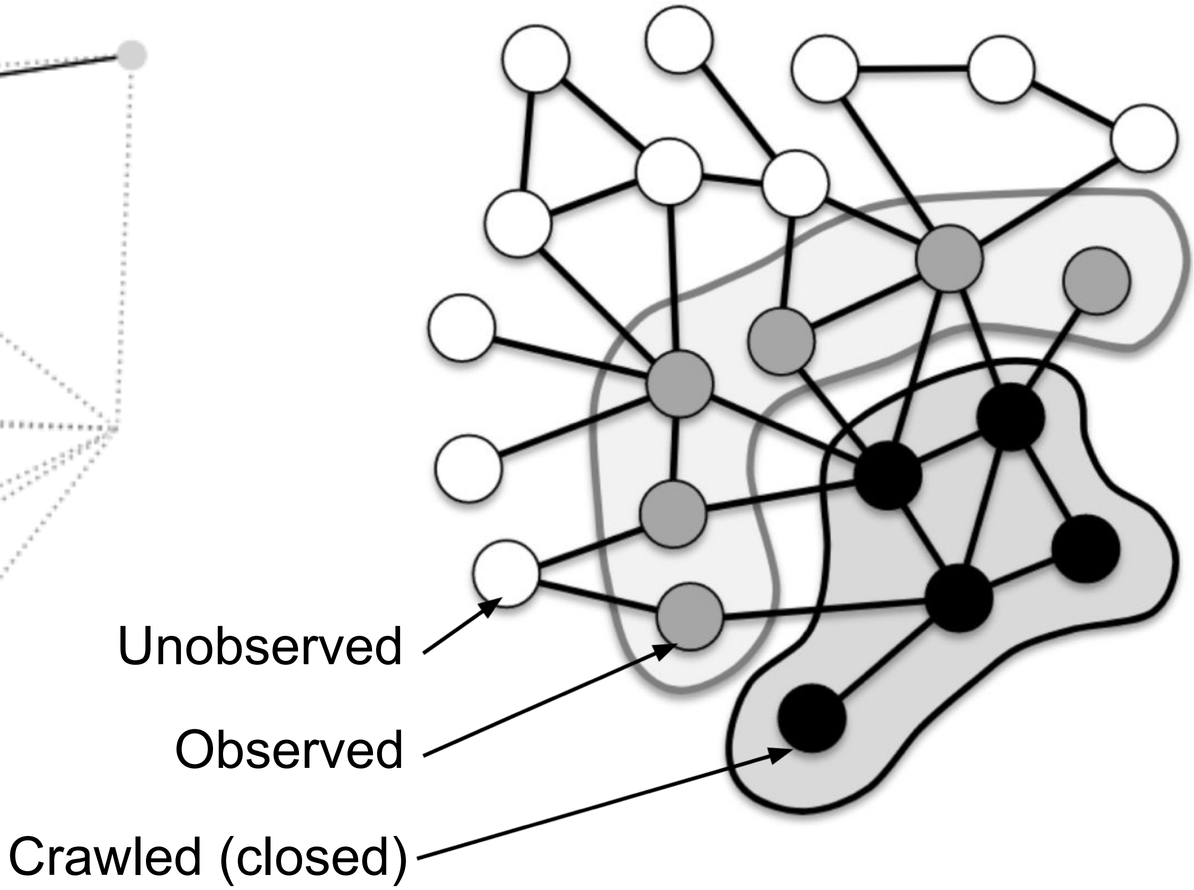
Edges represent the relationship (subscriptions, mentions, friendships)

- Opinion leaders
- Understanding trends
- Information spread

Importance = degree of vertex

How? Crawling process





Constraint: bandwidth limit of API

Statement of the problem

Let $G = (V, E)$ – static, undirected, unobserved graph

Budget of requests – n

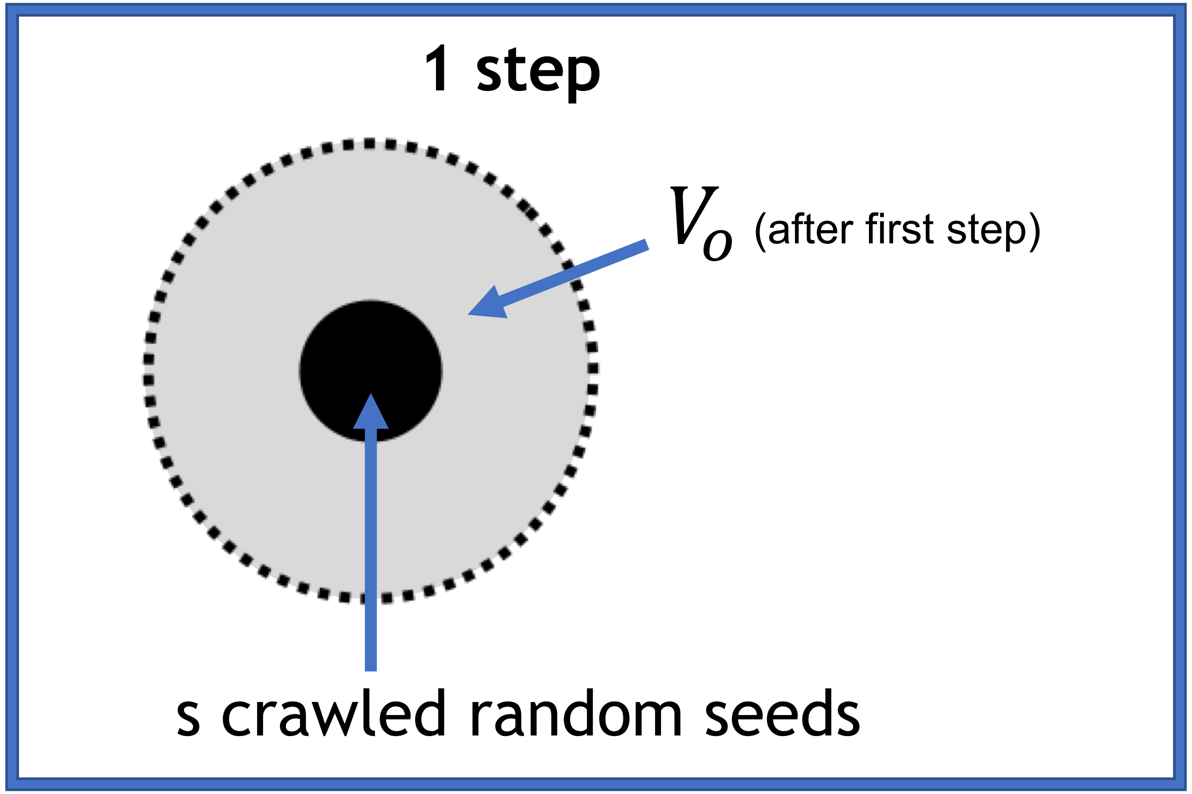
The goal is to detect top-p% degree nodes in the graph under a budget restriction

Propose 3-Step and 3-StepBatch to solve the problem

Description of proposed algorithms

input: s – number of random start seeds;
n – budget;
b – batch size (optional parameter, for 3-StepBatch only)

output: $p|V|$ nodes-candidates with highest degree

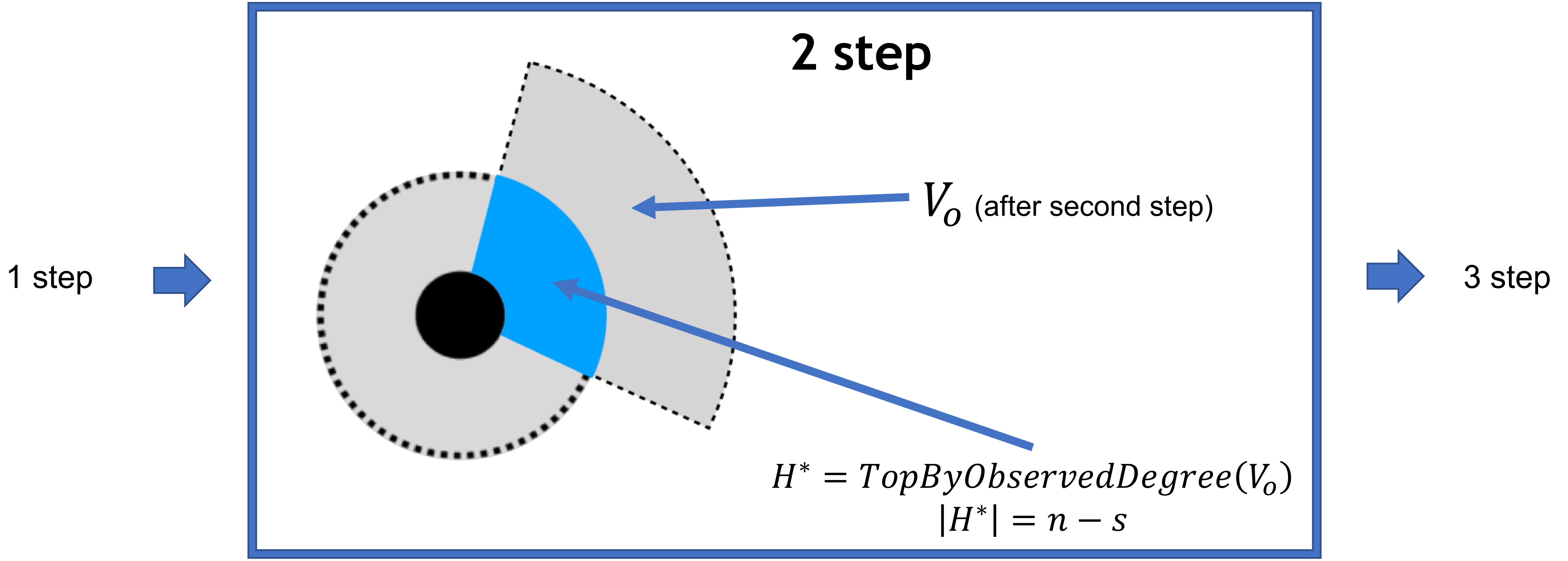


V_o – observed nodes

- Spends s requests on crawling of random vertices

Second step of 3-Step algorithm

- Ranks all currently observed nodes (V_o after first step) by their observed degree
- Selects top-$(n-s)$ nodes

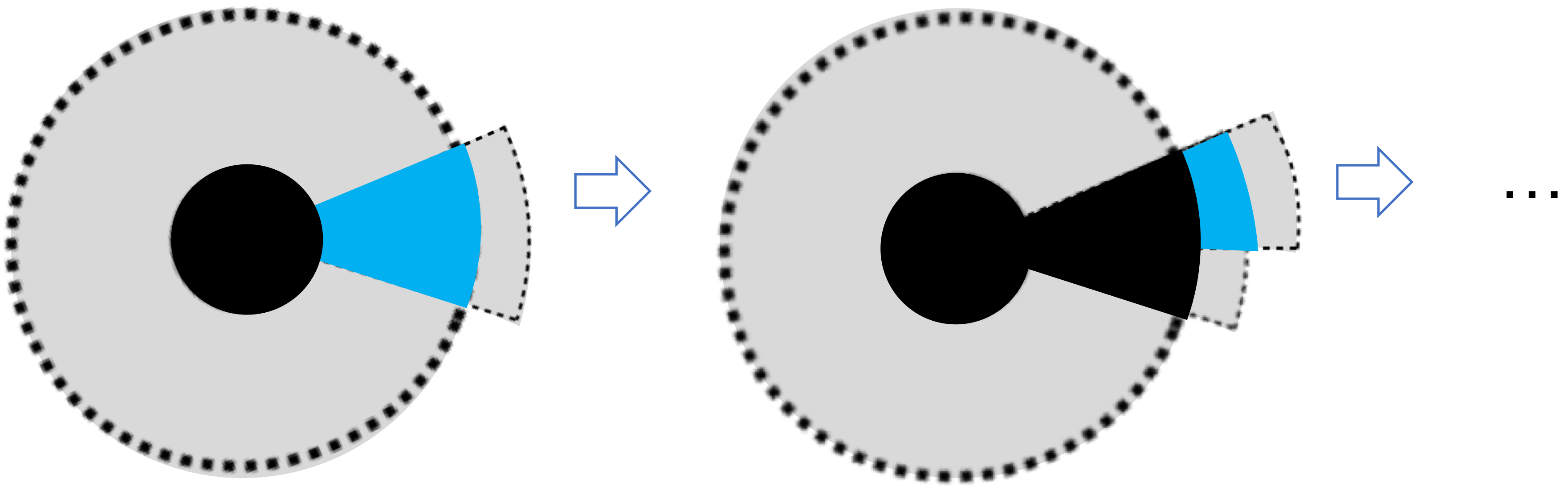


Second step of 3-StepBatch algorithm

1 step

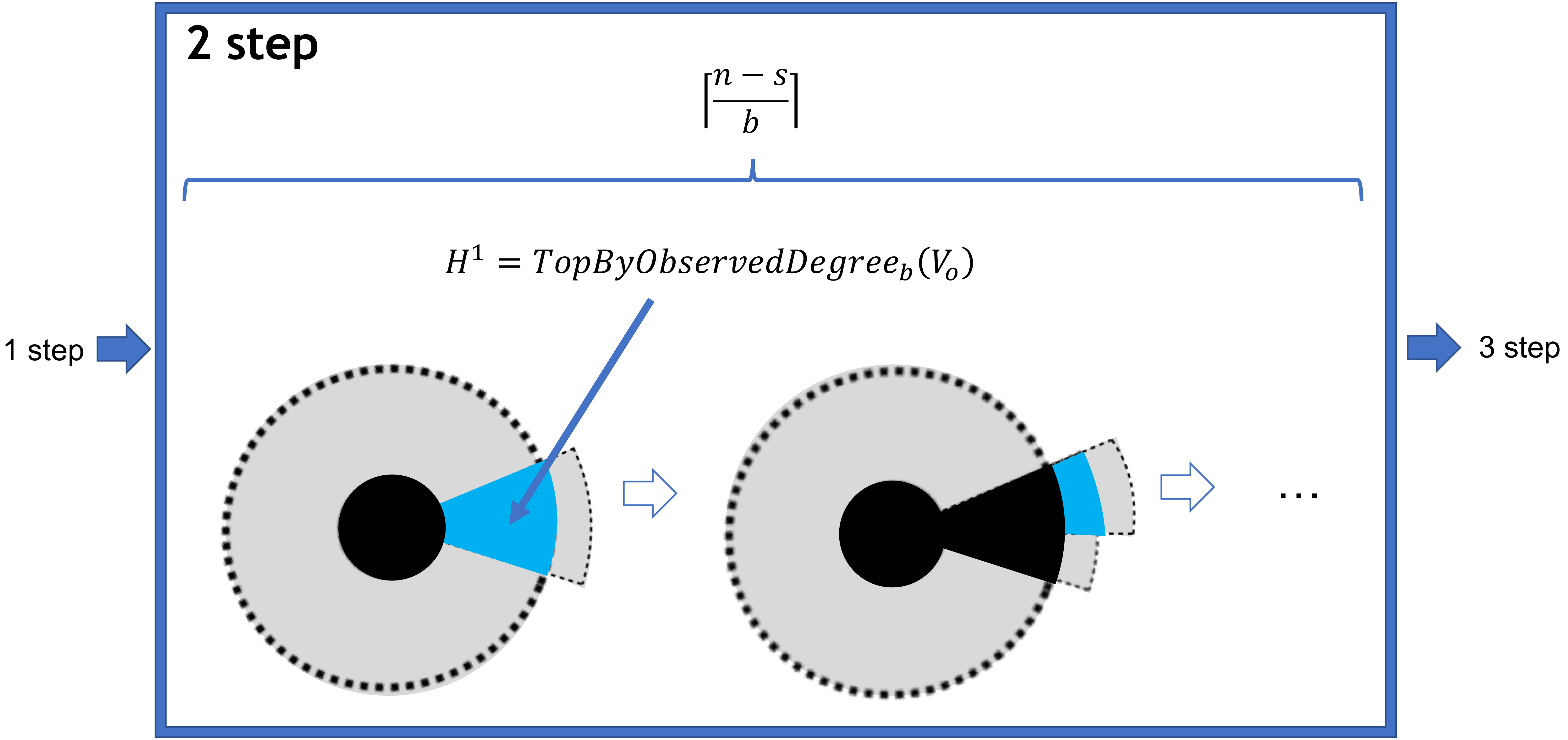
2 step

$$\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad}^{\left\lceil \frac{n-s}{b} \right\rceil}$$



3 step

Second step of 3-StepBatch algorithm



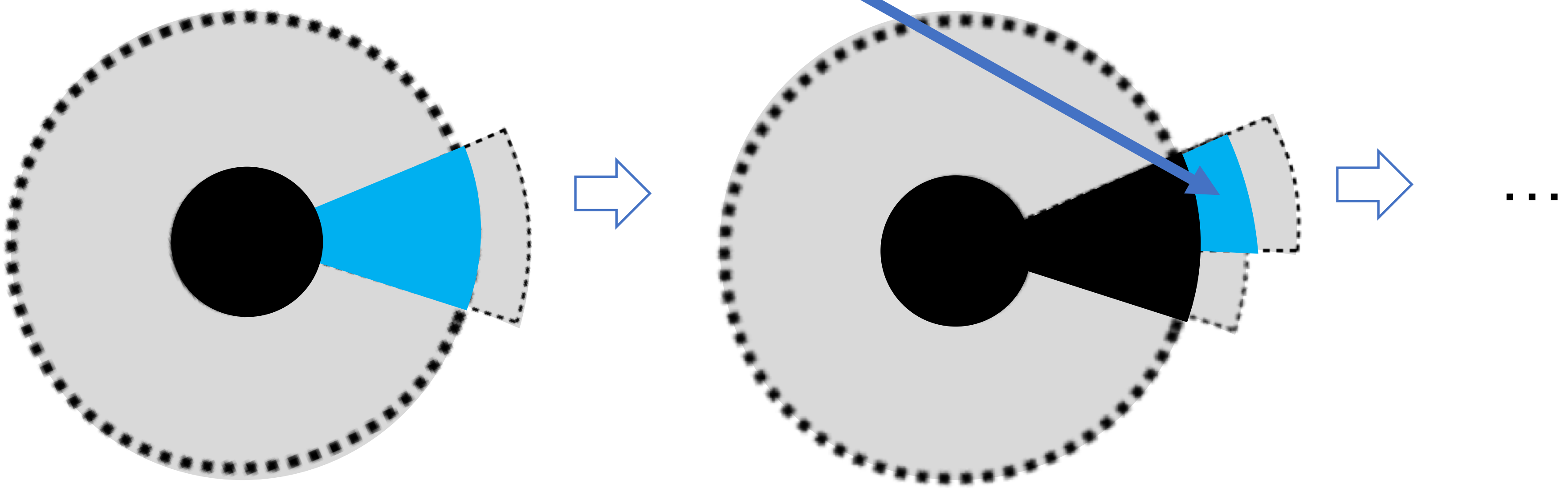
Second step of 3-StepBatch algorithm

1 step

2 step

$$\underbrace{\hspace{10em}}_{\left\lceil \frac{n-s}{b} \right\rceil}$$

$$H^2 = TopByObservedDegree_b(V_o)$$



3 step

Second step of 3-StepBatch algorithm

1 step

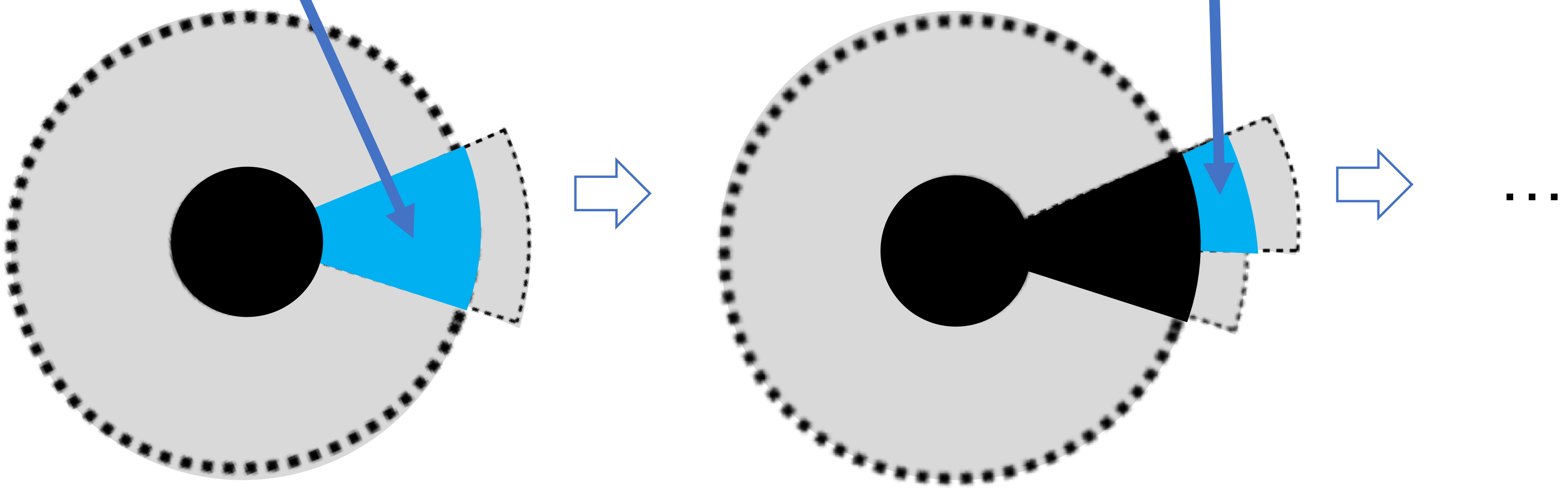
2 step

$$\underbrace{\qquad\qquad\qquad\qquad}_{}\left\lceil\frac{n-s}{b}\right\rceil$$

$$H^* = H^1 \cup H^2 \cup \ \dots \cup \ H^{\left\lceil\frac{n-s}{b}\right\rceil}$$

$$H^2 = TopByObservedDegree_b(V_o)$$

$$H^1 = TopByObservedDegree_b(V_o)$$

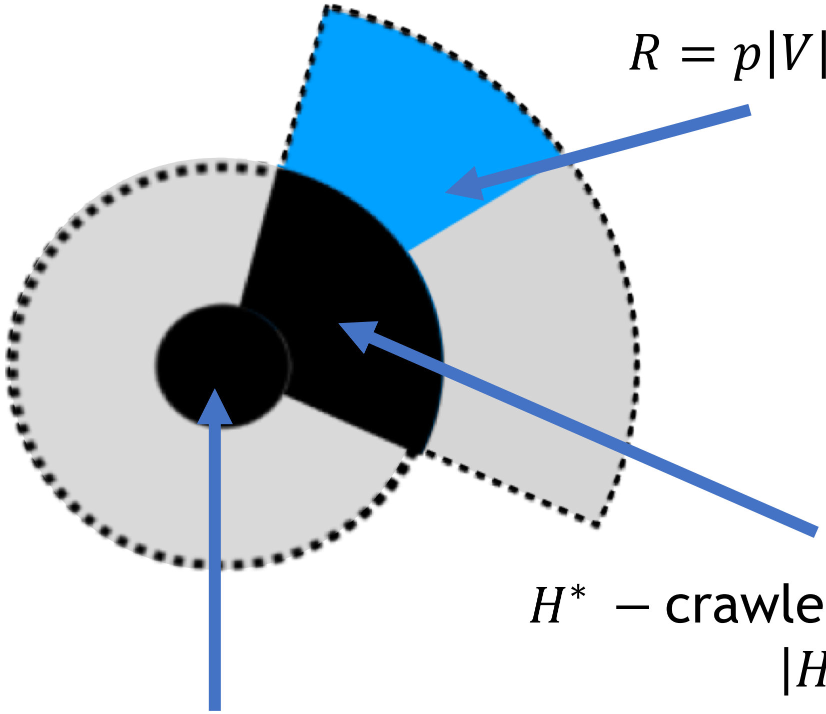


3 step

Third step of proposed algorithms

output: $p|\mathrm{V}|$ nodes-candidates with highest degree

3 step

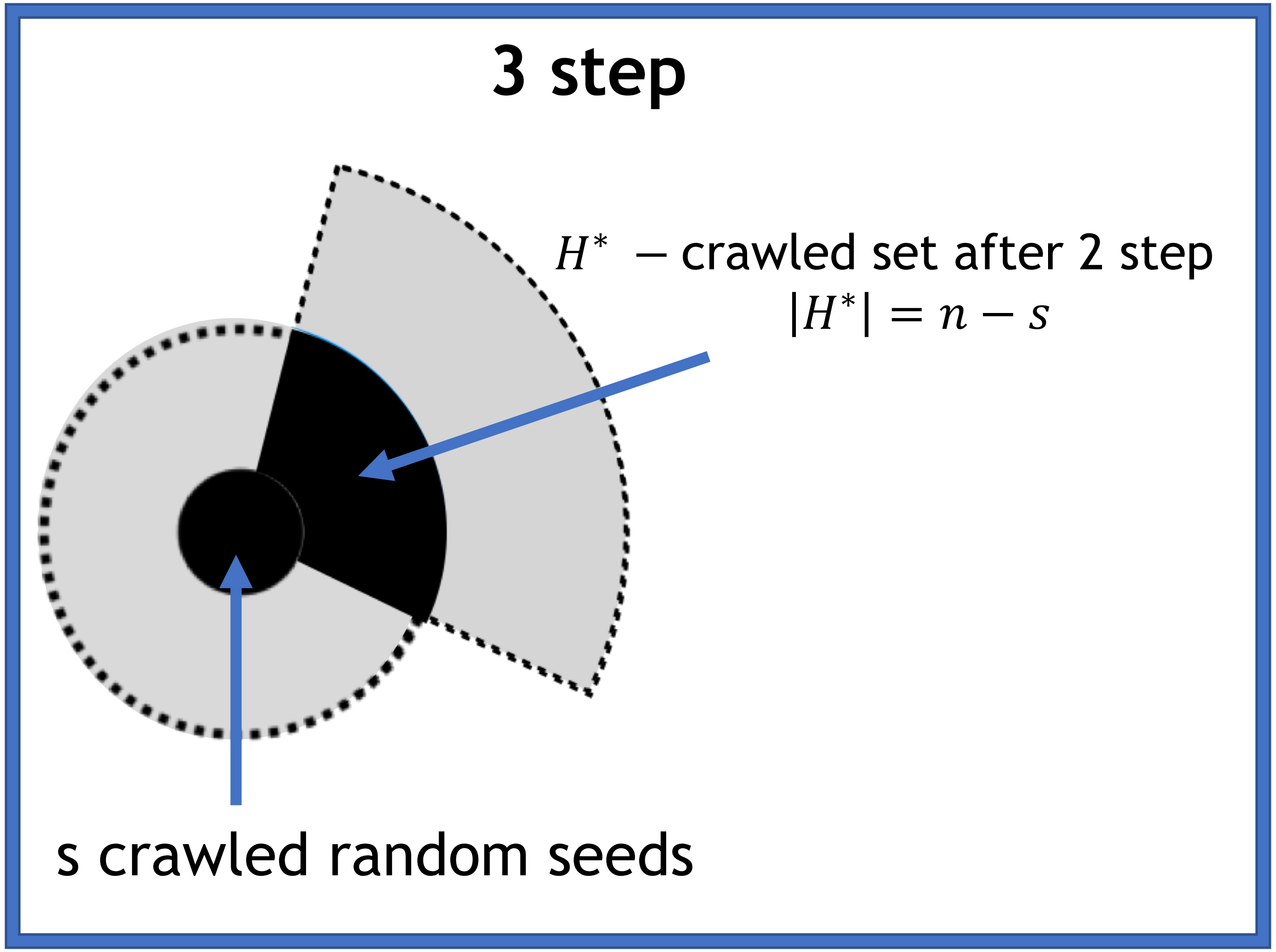


V_c – crawled set ($S \cup H^*$)

- If $p|\mathrm{V}| \geq n - s + ps$:
  $S^* = TopByDegree_{\lfloor ps \rfloor}(S)$
  $return\ \boldsymbol{A} = \boldsymbol{S}^* \cup \boldsymbol{H}^* \cup \boldsymbol{R}$

Third step of proposed algorithms

output: $p|V|$ nodes-candidates with highest degree



V_c – crawled set $(S \cup H^*)$

- If $p|V| < n - s + ps$:
  $return\ \boldsymbol{A} = \boldsymbol{TopByDegree_{p|V|}(V_c)}$

Quality measure and Optimal batch size

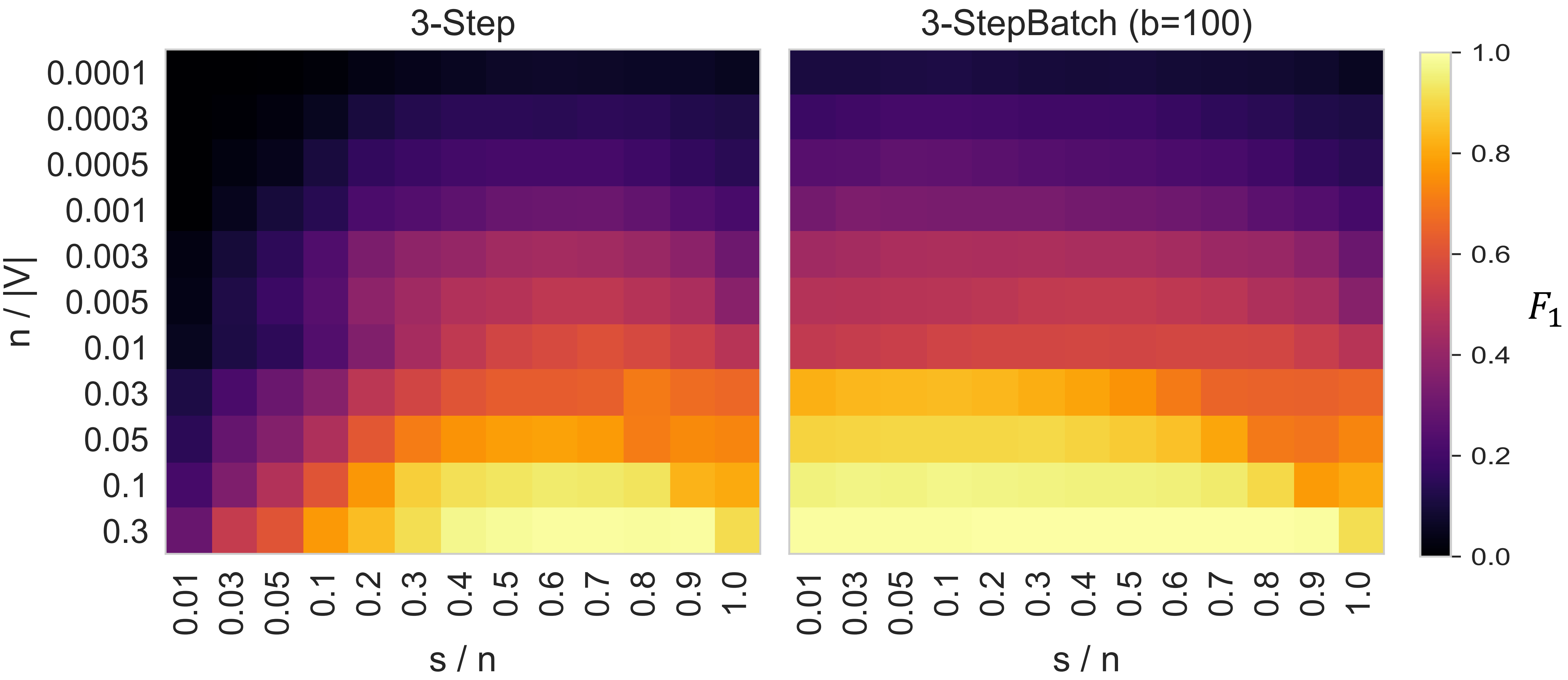
Quality measure = $\boldsymbol{F_1}$

Dataset: 24 samples of social networks of various sizes (see Appendix)

Optimal batch size:

- Vary b from 1 to 3000 nodes
- Vary s/n from 1% to 100%
- For two budget fractions, $n/|V|$: 5% and 0.5%
- $\boldsymbol{F_1}$ score for each combination was averaged across 24 graphs
- Optimal $\boldsymbol{b = 100}$

Optimal start seeds



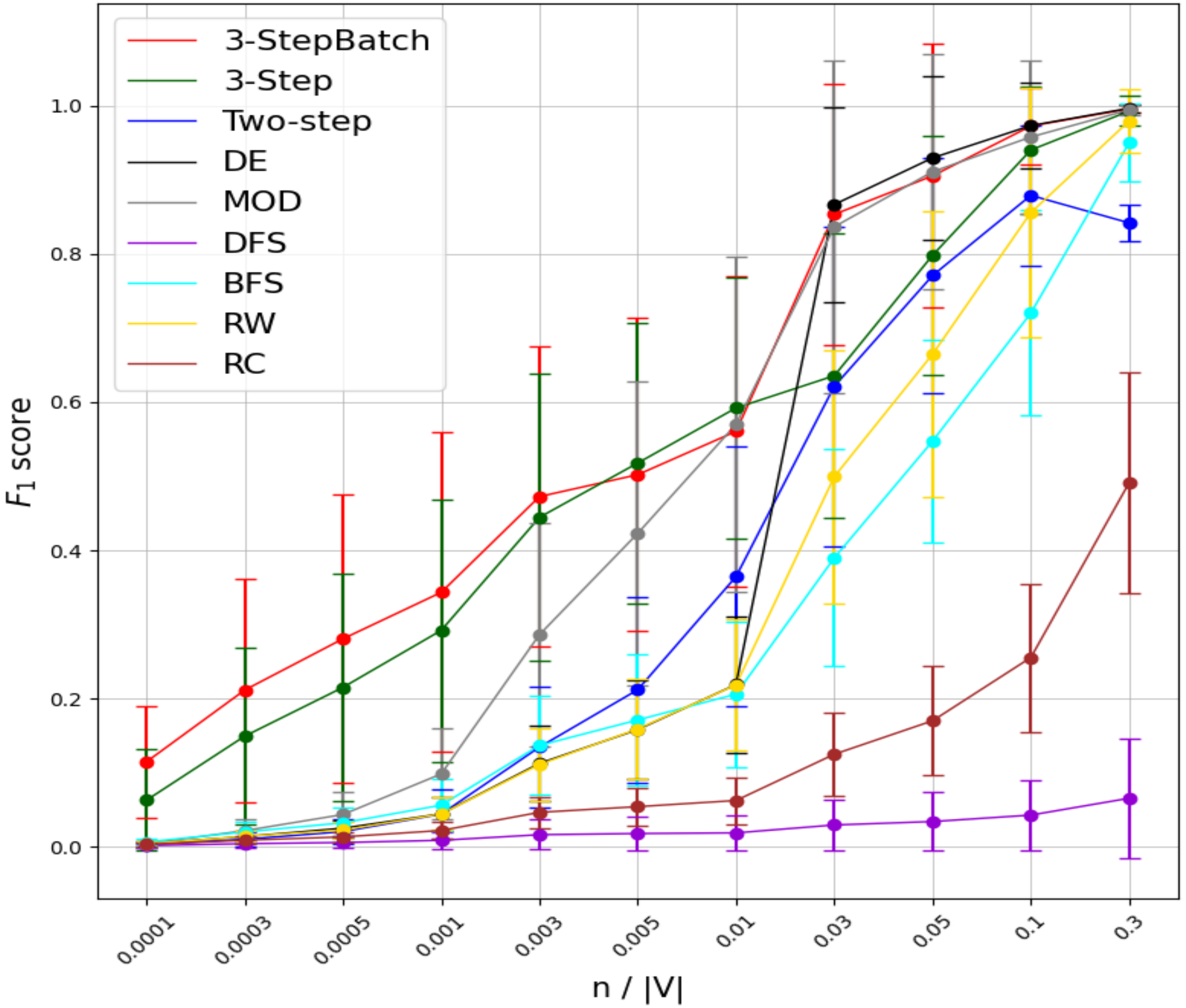
Optimal start seeds

Tab.1. Optimal number of of start random seeds s for 3-Step and 3-StepBatch algorithms, depending on the budget fraction n/N. Quality measure is averaged over 24 graphs.

	3-Step		**3-StepBatch**	
Budget, $\frac{n}{\lvert V\rvert}$	**Optimal** s	**Quality,** F_1	**Optimal** s	**Quality,** F_1
0.1%	$s = 0.6n$	0.30 ± 0.18	$s = 0.3n$	0.34 ± 0.21
0.3%	$s = 0.6n$	0.44 ± 0.19	$s = 0.1n$	0.46 ± 0.20
0.5%	$s = 0.6n$	0.51 ± 0.18	$s = 0.5n$	0.52 ± 0.17
1%	$s = 0.7n$	0.60 ± 0.16	$s = 0.7n$	0.57 ± 0.16
3%	$s = 0.8n$	0.71 ± 0.14	$s = 0.1n$	0.84 ± 0.19
5%	$s = 0.6n$	0.80 ± 0.16	$s = 0.3n$	0.91 ± 0.13
10%	$s = 0.6n$	0.94 ± 0.09	$s = 0.1n$	0.97 ± 0.06
30%	$s = 0.9n$	0.99 ± 0.01	$s = 0.4n$	0.99 ± 0.01

Comparison

ISP RAS



Alternative crawlers:

RC, MOD, RW, DFS, BFS,
Two-Stage Crawler[1], DE-Crawler[2].

Adapt crawlers to our task:

- While the number of steps is less than $p|V|$, the returned answer equals to all crawled nodes V_c;
- Further, the answer is defined as a subset of V_c of size $p|V|$ with highest degrees.

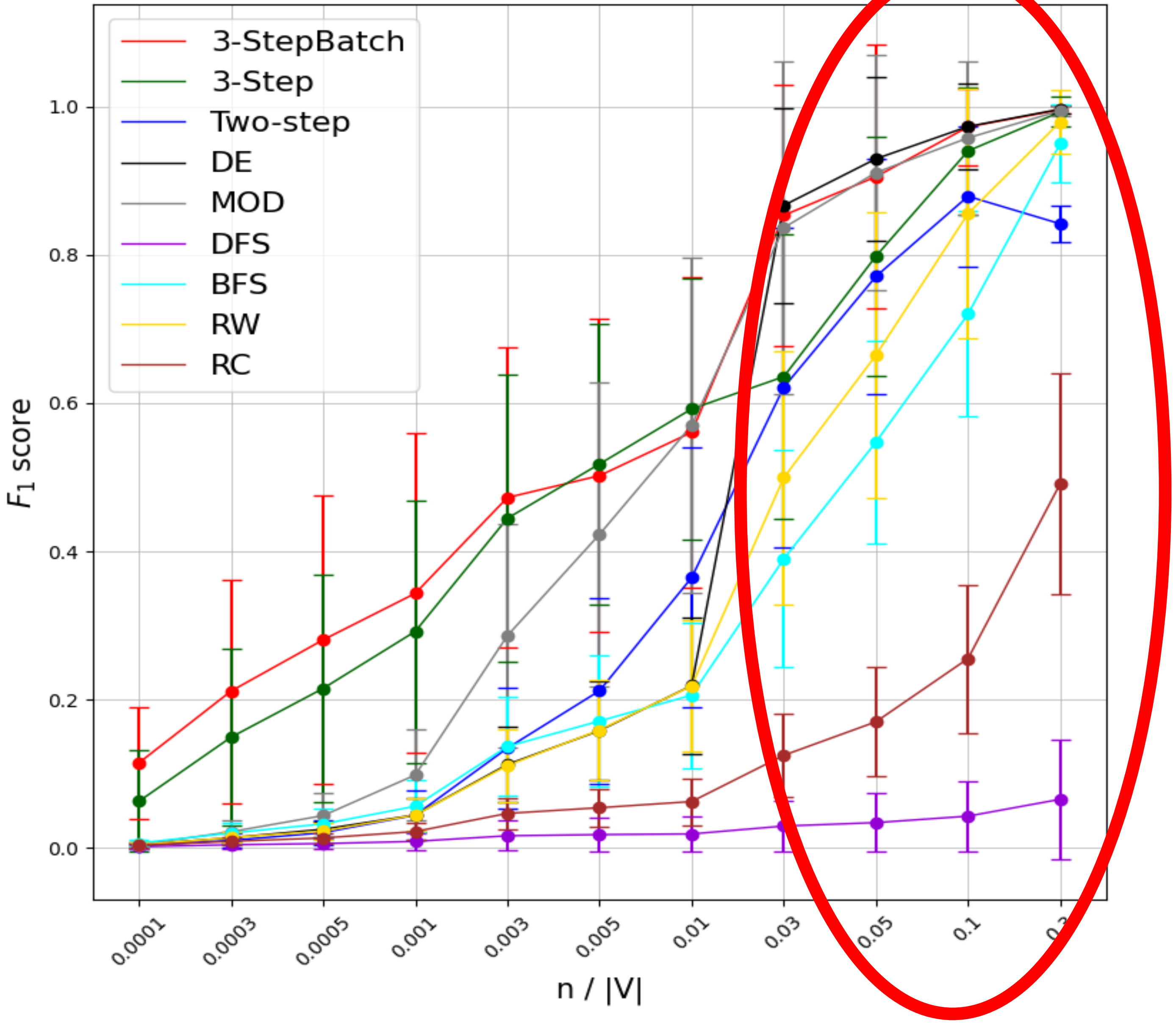
Use recommended parameters for algorithms:

for 3-Step – $s = 0.6n$;
for 3-StepBatch – $s = 0.1n, b = 100$;
for Two-Stage Crawler – $s = 0.5n$.

[1]Avrachenkov K., Litvak N., Prokhorenkova L. O., Suyargulova E. Quick detection of high-degree entities in large directed networks // 2014 IEEE International Conference on Data Mining / IEEE. 2014. P. 20–29.

[2]K. Areekijseree and S. Soundarajan, “De-crawler: A densification-expansion algorithm for online data collection,” in 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2018, pp. 164–169.

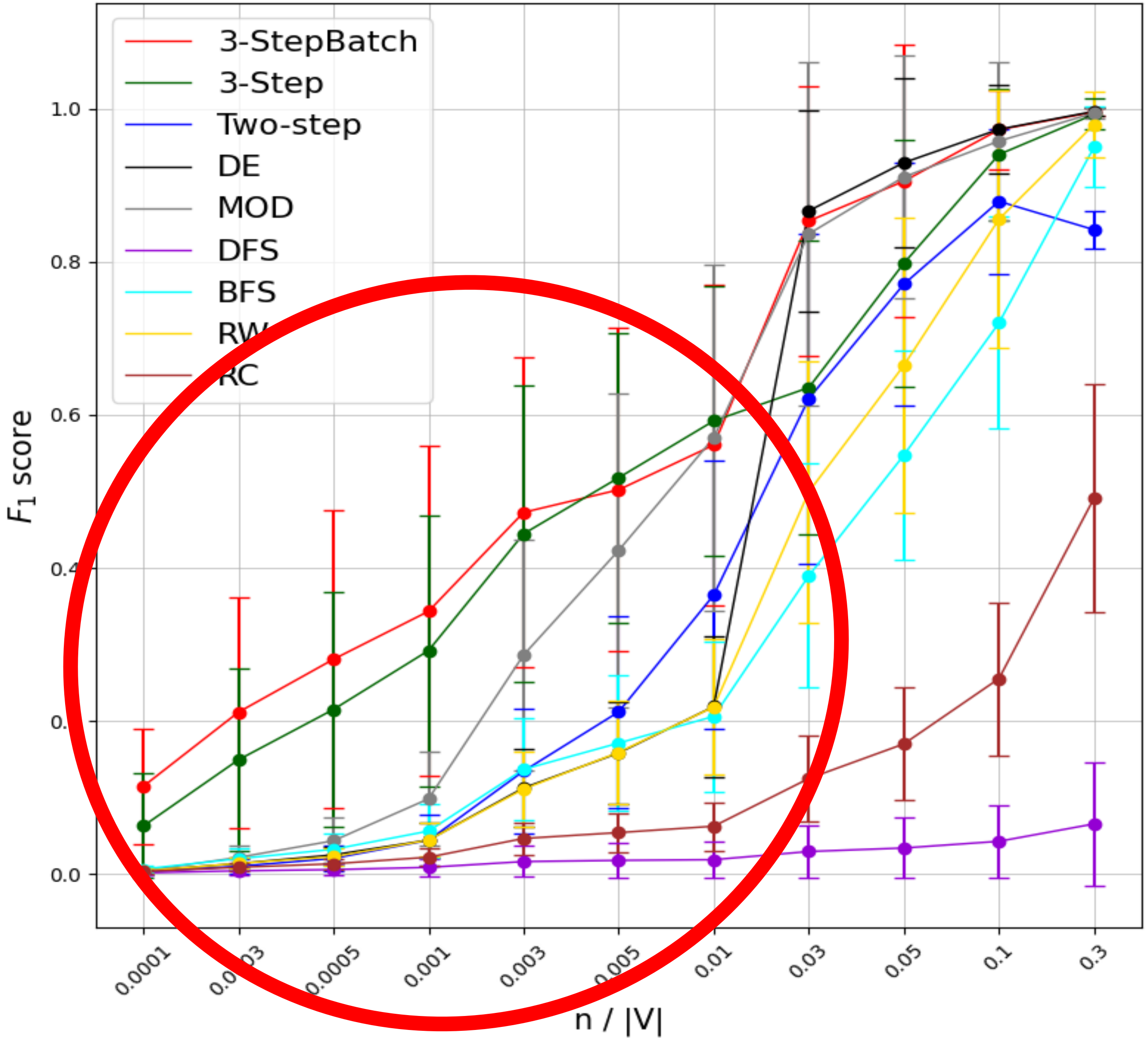
Comparison



Size of the target set – 1% of highest degree nodes

When $n \geq p|V|$,
the leading algorithms are 3-StepBatch, MOD and DE-Crawler

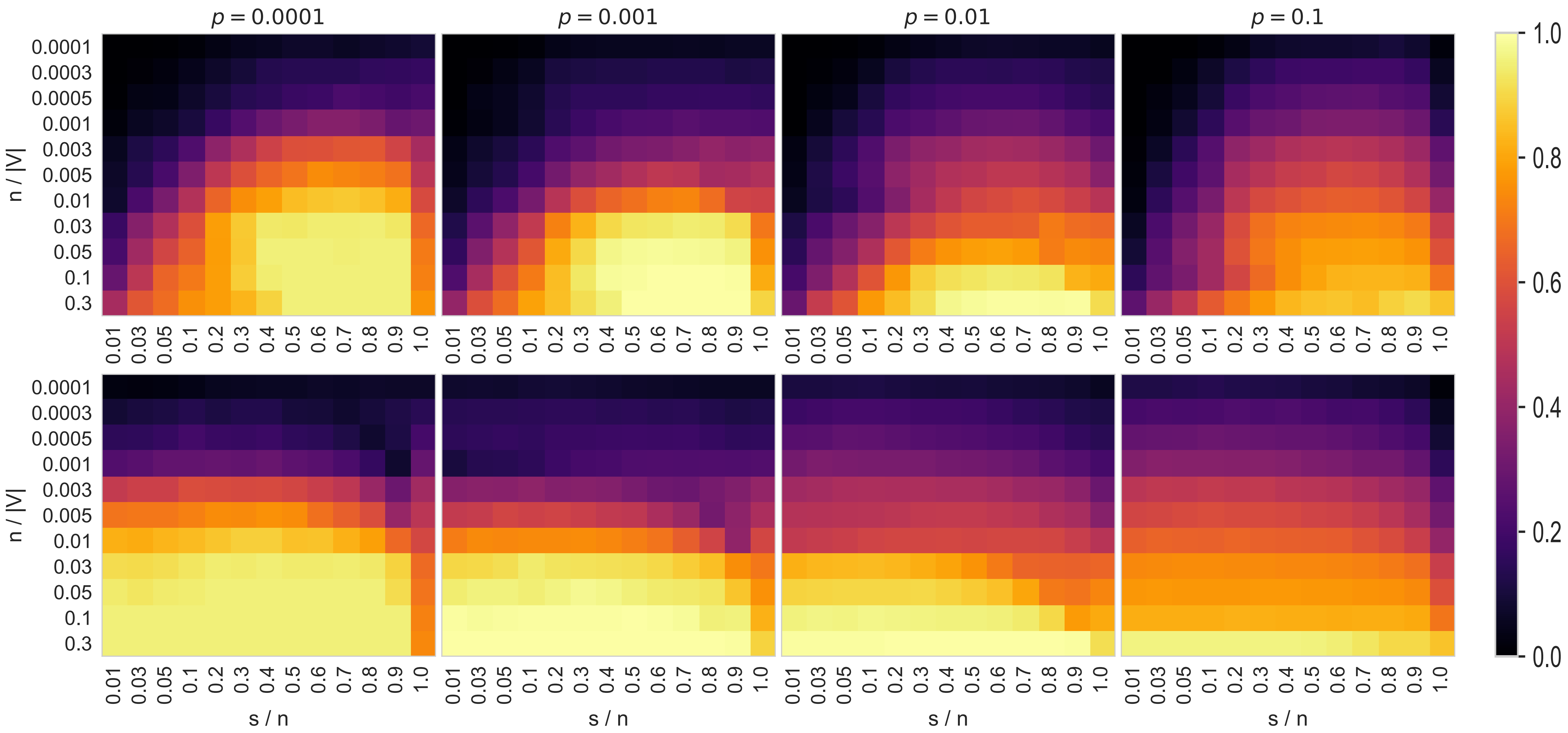
Comparison



Size of the target set – 1% of highest degree nodes.

When $n < p|V|$,
3-StepBatch and 3-Step algorithms are superior to other strategies

Variation of target set size



Conclusions

- ☑ Proposed 3-Step and 3-StepBatch algorithms for quick detection top-p% by degree nodes

- ☑ The comparison results showed that 3-StepBatch is no worse than the alternatives:
  - $n \geq p|V|$: has similar result with alternatives
  - $n < p|V|$: outperforms the alternatives

- ☑ To detect top-1% of hubs with 90% precision, one needs to crawl 5% of graph nodes in average with 3-StepBatch algorithm

- ☑ Implemented them in framework https://crawling-framework.github.io/

Thank you for your attention!

Danil Shaikhelislamov[1], Mikhail Drobyshevskiy[2], Denis Turdakov[2,3],
Alexander Yatskov[2], Maksim Varlamov[2], Denis Aivazov[1,2]

[1]Moscow Institute of Physics and Technology (State University), Moscow, Russia
[2]Ivannikov Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia
[3]Lomonosov Moscow State University, Moscow, Russia

IVANNIKOV MEMORIAL WORKSHOP, OREL
SEPTEMBER 25-26, 2020

Appendix. Dataset

ISP RAS

Network	Nodes	Edges
petster-hamster [2]	2 000	16 098
socfb-Bingham82 [2]	10 001	362 892
soc-anybeat [1]	12 645	49 132
ego-gplus [2]	23 613	39 182
socfb-Penn94 [1]	41 536	1 362 220
slashdot-threads [2]	51 083	116 573
loc-brightkite_edges [2]	56 739	212 945
soc-brightkite [1]	56 739	212 945
socfb-wosn-friends [1]	63 392	816 886
soc-themarker [1]	69 317	1 644 794
soc-slashdot [1]	70 068	358 647
soc-BlogCatalog [1]	88 784	2 093 195
livemocha [2]	104 103	2 193 083
epinions [2]	119 130	704 267
petster-friendships-cat [2]	148 826	5 447 464
douban [2]	154 908	327 162
digg-friends [2]	261 489	1 536 577
soc-twitter-follows [1]	404 719	713 319
petster-friendships-dog [2]	426 485	8 543 321
munmun_twitter_social [2]	465 017	833 540
com-youtube [2]	1 134 890	2 987 624
soc-pokec-relationships [2]	1 632 803	22 301 964
flixster [2]	2 523 386	7 918 801
youtube-u-growth [2]	3 216 075	9 369 874

Tab.2.
24 Real-world social graphs used in experiments.

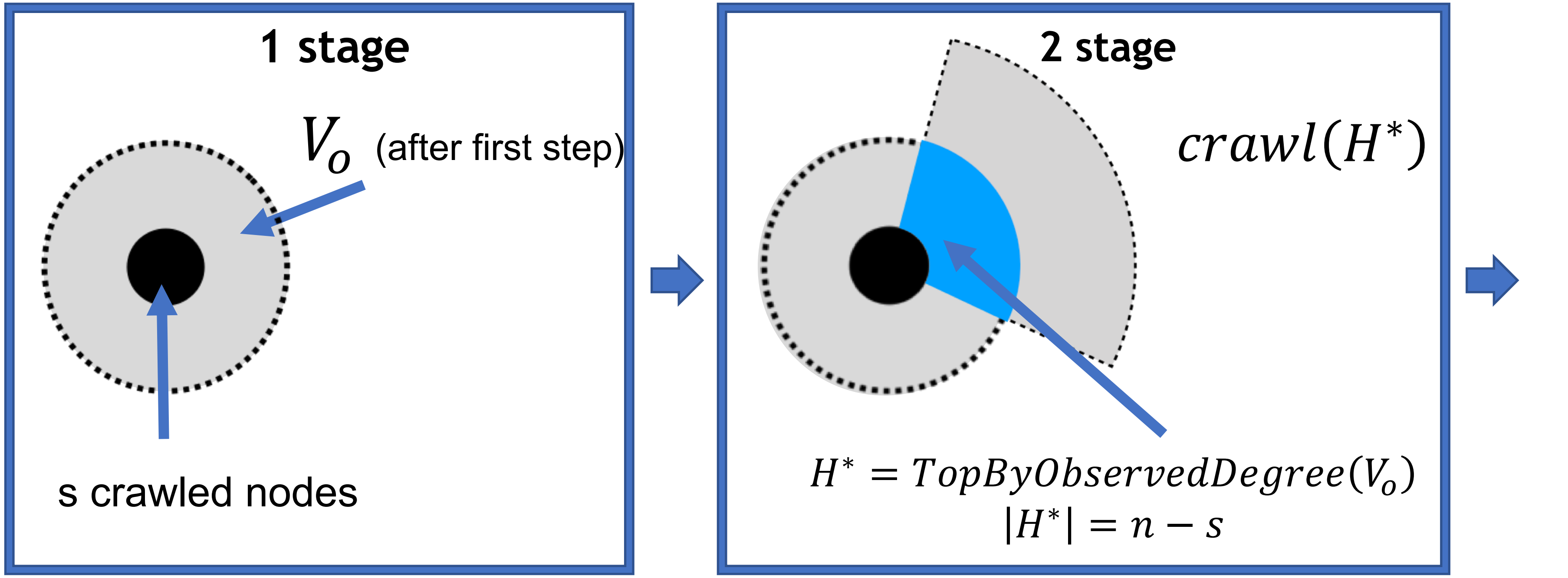
[1] R. A. Rossi and N. K. Ahmed, “The network data repository with interactive graph analytics and visualization,” in AAAI, 2015. [Online].
Available: http://networkrepository.com

[2] J. Kunegis, “The koblenz network collection,” in Proc. Int. Conf. on World Wide Web Companion, 2013, pp. 1343–1350. [Online].
Available: http://konect.cc/networks/

Appendix. Two-stage algorithm *

input: s (300-500) – number of random start nodes;
n (~1000) – budget

output: k (50-250) nodes-candidates with high degree;
$k < n$



$return\ Top_k\ (V_c)$

*Avrachenkov K., Litvak N., Prokhorenkova L. O., Suyargulova E. Quick detection of high-degree entities in large directed networks // 2014 IEEE International Conference on Data Mining / IEEE. 2014. P. 20–29.