

# Supplementary Material For Modeling the Global Citation Network using the Scalable Agent-based Simulator for Citation Analysis with Recency-emphasized Sampling (SASCA-ReS)

Minhyuk Park<sup>1</sup>, Haotian Yi<sup>1</sup>, Tandy Warnow<sup>1</sup>, George Chacko<sup>1,2\*</sup>

<sup>1\*</sup>Siebel School of Computing and Data Science, University of Illinois  
Urbana-Champaign, 201 N. Goodwin Ave, Urbana, 21801, IL, USA.

<sup>2</sup>Grainger College of Engineering, University of Illinois  
Urbana-Champaign, 1308 W Green St, Urbana, 61801, IL, USA.

\*Corresponding author(s). E-mail(s): [chackoge@illinois.edu](mailto:chackoge@illinois.edu);  
Contributing authors: [minhyuk2@illinois.edu](mailto:minhyuk2@illinois.edu); [yi54@illinois.edu](mailto:yi54@illinois.edu);  
[warnow@illinois.edu](mailto:warnow@illinois.edu);

## Contents

<b>S1 Additional details for SASCA-ReS</b>	<b>3</b>
S1.1 Description of PyABM . . . . .	3
S1.2 Description of SASCA-ReS . . . . .	4
<b>S2 Figures</b>	<b>7</b>
<b>S3 Tables</b>	<b>8</b>
<b>S4 Computing a SASCA-ReS network with 218 million nodes</b>	<b>10</b>
<b>S5 Additional Details for Materials and Methods</b>	<b>11</b>

## List of Figures

S1 Impurity-based feature importances for abm14 . . . . .	7
---	---

## List of Tables

S1	In-degree statistics and average local clustering coefficients for agents in abm14 variants . . . . .	8
S2	Disruption in mavericks . . . . .	9

## S1 Additional details for SASCA-ReS

The SASCA-ReS algorithm is an extension of agent-based modeling code developed by us to simulate the growth of citation networks. The major releases in this development path are v3 (PyABM) [1] written in Python, v5 (SASCA and SASCA-S) [2], written in C++, and v6 (SASCA-ReS) [3], written in C++.

SASCA-ReS differs from SASCA-s and PyABM in several respects, some of which are about computational performance and others about the model. Here we discuss the difference in terms of the model. Since the details of the PyABM model are available [1], we provide first a description of PyABM and then specify how SASCA-ReS differs from it.

### S1.1 Description of PyABM

The PyABM model can be described as. having 5 basic steps, as provided in the paper on PyABM [1]:

1. The network begins with a seed network with nodes and directed edges. Each year, a batch of new agents is created, defined by the rate of agent birth (typically 3% to 10%, a parameter given to the simulator).
2. To generate a new agent, a node is selected from the network at random and designated as a “generator”; this generator then spawns the new agent, and the new agent cites the generator. The new agent is also assigned a fitness value drawn from a power-law distribution which does not change over time.
3. The newly created agent is assigned an integer-valued quota of references it can make to other nodes, where the quota is drawn from a distribution derived from real-world citation data. One of these citations is to its generator, and for a subset of the new agents (typically 12% of all new agents), an additional citation is made to another a node from its own year. The remainder of the citations are made to other nodes in the network.
4. For each agent, the selection of which citations to make depends on the target nodes properties, namely: fitness, current in-degree, and year of publication, as well as the agent’s own relative preference for these properties, which we refer to as its “phenotype”. The agent also has a parameter, which we will refer to here as  $\alpha_P$  (where “P” refers to PyABM), that determines the fraction of its references that will come from the 1-hop neighborhood of its generator. Each node in the network is a potential recipient of a citation, and hence is scored according to the agent’s phenotype. Then, given the number of citations the agent should make to nodes one edge from its generator, the agent selects the nodes to cite based on a weighted sampling function using the A-res algorithm presented in [4], and the same process is used for selecting nodes to cite that are further than one edge from the generator.
5. Once an agent exhausts its quota of references, it does not make any more citations and can only receive citations from other agents.

Some comments are relevant. Note that the agent’s phenotype determines the relative importance of fitness, in-degree (which determines the preferential attachment score), year of publication, and whether a node is in the 1-hop neighborhood of its generator

(i.e., all the nodes of distance exactly one edge from the generator). For example, when fitness and preferential attachment are equally weighted, then the assigned fitness of a node drives the early period of citation accumulation, and gradually decreases as the node accumulates in-degree. The process ensures that nodes with higher scores are more likely to be selected while maintaining stochasticity.

## S1.2 Description of SASCA-ReS

The model for SASCA-ReS is nearly identical to PyABM. Steps 1–3 and 5 are the same as for PyABM, which means that every node is spawned by a randomly selected generator, which it cites; every node has a fitness value; and a small subset of the new nodes in each year will cite nodes within the same year. There are however substantial changes to Step 4, which determines the remaining citations an agent will make in order to fill its quota. As with PyABM, the distance to the generator impacts this decision, and we define the 2-hop neighborhood of the generator to be all those nodes of distance 1 or 2 to the generator, and hence the 1-hop neighborhood is the set of nodes of distance exactly 1 from the generator; thus, the 2-hop neighborhood contains the 1-hop neighborhood. To define how SASCA-ReS decides what other node to cite, we use the following parameters:

- $q$  (quota): the allotted quota of citations to be made (out-degree)
- $IF$  (idiosyncratic fraction): the fraction of the quota that is allotted to idiosyncratic citations (default  $IF = 5\%$ )
- $\alpha$ : the fraction of the number of citations made within the 2-hop neighborhood of the generator that are made to citations in the 1-hop neighborhood (no default)
- $SY$ : the fraction of agents in a given year that make citations to other agents from the same year (default  $SY = 12\%$ )
- $B$ : the maximum number of nodes that each agent scores within the 1-hop and 2-hop neighborhoods of the generator node (default  $B = 10,000$ )

We now describe how SASCA-ReS uses these parameters. One important aspect of the changes in Step 4 is that SASCA-ReS only scores at most  $B$  node in the 1-hop neighborhood and  $B$  nodes in the 2-hop neighborhood; an agent may cite a node that is not in the 2-hop neighborhood, but if so then such citations are made randomly and do not require scoring the node (furthermore, we call such citations “idiosyncratic”). The number of idiosyncratic citations it will make is  $\lfloor q \times IF \rfloor$ , and so can be 0 if  $q < IF^{-1}$ . The citations going to the 2-hop neighborhood is divided into those going into the 1-hop and those going to nodes of distance 2 from the generator node, with  $\alpha$  defining the fraction of the total that go to the 1-hop. proportion going to the 1-hop. Thus, when  $\alpha = 0$ , all these citations go to the 1-hop neighborhood, and when  $\alpha = 1$ , then all these citations go to nodes at distance 2 from the generator.

There is also one more very important difference, which has to do with how SASCA-ReS ensures a better fit to the recency distribution: it bins all the nodes at distance 1 from the generator ( $N_1$ ) as well as the nodes that are at distance 2 from the generator ( $N_2$ ), by year of publication into a small number of bins. The first ten bins are for the first ten years, as follows. Assuming the agent is created in year  $y$ , then papers in  $N_1$  that were published in the previous year are in  $Bin_1$ , papers in the year before (i.e.,

year  $y-2$ ) are in  $Bin_2$ , and so forth, producing bins  $Bin_1, Bin_2, \dots, Bin_{10}$ . After this, the next eight bins are for five years at a time (i.e.,  $Bin_{11}$  has years 11 – 15,  $Bin_{12}$  has years 16 – 20, etc.). Finally, all publications (nodes) that are 51 years or older at the time of the agent’s creation are in the final bin. Note that the nodes in the final bin are also not scored, just like idiosyncratic citations, and instead selected randomly for citations. The same binning is done for the nodes in  $N_2$ , producing bins  $Bin'_1, Bin'_2$ , etc. Each agent then determines the integer count of how many citations to make within each bin based on a computed “recency table” from a real-world network, which reflects the preference to cite publications from recent years more than papers many years ago.

We first process  $N_1$  and then process  $N_2$ . Each bin is processed in turn, starting with  $Bin_1$  and then processing  $Bin_2$ , etc., for  $N_1$ , and a similar sequence for  $N_2$ . This protocol may succeed in achieving the quota for the agent. However, if the number of nodes in a given bin is not large enough to achieve the target number of citations from that bin, then the process may not meet the quota entirely. In this case, we perform a post-processing step to make additional citations, as follows. For a given bin that was insufficient in size to meet its quota, we check the bins already processed (i.e., bins with lower index), one by one, to see if any nodes were left un-cited by the agent. If so, we can make a citation to any such nodes. We repeat this until we meet the quota. If this is still insufficient, we then move on to the higher indexed bins. Finally, if the 2-hop neighborhood is too small to cover the quota, then we perform additional idiosyncratic citations, which do not involve scoring.

Note that because of this binning property, the phenotype for each agent no longer considers recency in calculating the score of a target node, and instead only considers the relative importance of preferential attachment and fitness. This relative importance is specified by assigning weights between 0 and 1 for these two properties that sum to 1, with the obvious interpretation that whichever has a higher weight will contribute more to the score. For example, if the weight for preferential attachment is 1 and the weight for fitness is 0, then fitness is not considered at all in computing the weight of any target node, but if both are weighted 0.5 then they both contribute equally to the weight.

We now summarize this discussion and provide some details for how nodes are selected to be cited. Recall that the parameters we use specify a fraction of the new agents in a single year will perform a single same-year citation; this is done at the beginning of each year of the simulation. For a new agent with a given integer-valued quota  $q$  and generator node  $x$ , the steps for fulfilling the quota is as follows.

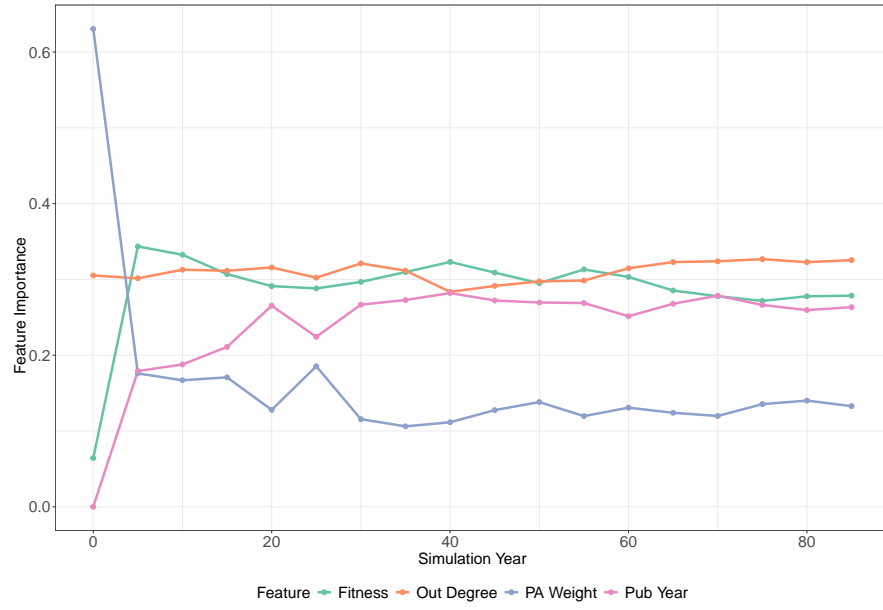
1. Make  $\lfloor IF \times q \rfloor$  idiosyncratic citations and decrement  $q$  by this value.
2. Cite the generator node  $x$  and decrement  $q$ .
3. If the agent was selected to make a same-year citation, then select a random other agent from the same year that was *not* selected for same-year citations and cite it; then decrement  $q$ .
4. Obtain  $N_1$  and  $N_2$ , where  $N_i$  is the set of nodes that are distance  $i$  from the generator node  $x$ .
5. Let  $q_1 := \lfloor \alpha q \rfloor$  and let  $q_2 := q - q_1$

6. Randomly select  $\min\{|N_1|, 10000\}$  nodes from  $N_1$  and score them according to the agent’s phenotype. Cite  $q_1$  nodes from the selected set based on their scores for all bins except last. Cite randomly from the last bin.
7. Randomly select  $\min\{|N_2|, 10000\}$  nodes from  $N_2$  and score them according to the agent’s phenotype. Cite  $q_2$  nodes from the selected set based on their scores for all bins except last. Cite randomly from the last bin.
8. Make additional idiosyncratic citations if needed to satisfy quota.

***The “no- $\alpha$ ” variant***

Recall that parameter  $\alpha$  determines the proportion of citations that should go to  $N_1$  out of the 2-hop neighborhood, with higher  $\alpha$  values leading to more citations from  $N_1$ . We also allow for a “no- $\alpha$ ” variant, which has the consequence that instead of partitioning the 2-hop neighborhood into  $N_1$  and  $N_2$ , we just sample from the 2-hop neighborhood, which is  $N_1 \cup N_2$ . As with the default way of running SASCA-ReS, we partition  $N$  into bins based on year of publication, and scoring and selection of nodes to cite follows the same structure.

## S2 Figures



**Fig. S1 Impurity-based feature importances for abm14** Here we show the impurity-based feature importances of random forest regressors trained on abm14 datasets at different years of the simulation. Each random forest regressor consisted of 100 decision trees and predicts the final citation received by an agent using four features: fitness, out\_degree, preferential attachment weight, and publication year.

## S3 Tables

abm14 variants	alcc	min	25th	median	75th	max
ra	0.0839	0	5	12	24	260404
ra_noalpha	0.0239	0	3	7	14	329261
alpha0.05	0.0212	0	3	7	15	302766
alpha0.95	0.1498	0	8	20	39	417066
pa0.05	0.0824	0	5	12	26	342999
pa0.95	0.0883	0	4	11	22	443169
alpha0.00	0.0054	0	3	7	14	338176
alpha0.20	0.0530	0	4	9	18	244258
alpha0.40	0.0807	0	4	11	23	341497
alpha0.60	0.1052	0	6	14	28	311606
alpha0.80	0.1296	0	7	17	34	372994
alpha1.00	0.1605	0	8	21	42	437329

**Table S1 In\_degree statistics and average local clustering coefficient (alcc) for agents in different simulated networks with 14 million nodes.** The networks are variants of the abm14 network produced by varying whether  $\alpha$  and preferential attachment weight are randomized or static. For each network we display statistics for its in\_degree distribution and alcc. Note that high  $\alpha$  settings increase in\_degree and alcc of agent nodes.



	condition	maverick type	avg_n_i	avg_n_j	avg_n_k
1	od249_f1	maximizer	92984.67	67828.67	262947.00
2	od249_f1	randomnik	30.67	32.00	3269.33
3	od249_f1	minimizer	2.33	10.00	2023.67
4	od249_f1	control	8.33	22.00	10355.33
5	od249_f10	maximizer	99286.67	80901.33	285889.00
6	od249_f10	randomnik	380.67	92.67	3208.67
7	od249_f10	minimizer	395.00	88.67	2013.00
8	od249_f10	control	971.00	567.33	15230.00
9	od249_f100	maximizer	136190.00	123351.33	251704.33
10	od249_f100	randomnik	9418.00	291.00	3527.67
11	od249_f100	minimizer	5916.33	207.33	1963.33
12	od249_f100	control	10927.33	944.33	6683.00
13	od249_f1000	maximizer	155252.67	114938.33	230361.67
14	od249_f1000	randomnik	12805.33	330.33	2884.00
15	od249_f1000	minimizer	10475.00	262.33	1941.67
16	od249_f1000	control	18705.33	6958.67	42342.67
17	od10_f1	maximizer	118.33	42.33	6566.67
18	od10_f1	randomnik	2.33	0.67	180.00
19	od10_f1	minimizer	2.33	0.67	110.67
20	od10_f1	control	2.67	0.33	167.33
21	od10_f10	maximizer	2123.00	380.33	21042.00
22	od10_f10	randomnik	18.67	3.00	106.00
23	od10_f10	minimizer	15.00	3.00	76.67
24	od10_f10	control	147.33	119.33	5671.00
25	od10_f100	maximizer	29772.33	921.33	8540.00
26	od10_f100	randomnik	1576.00	16.67	239.33
27	od10_f100	minimizer	575.33	12.00	77.67
28	od10_f100	control	6774.67	335.33	5716.33
29	od10_f1000	maximizer	63427.00	1069.33	10041.00
30	od10_f1000	randomnik	845.00	11.00	134.67
31	od10_f1000	minimizer	444.00	9.67	85.67
32	od10_f1000	control	13624.33	472.67	4918.00

**Table S2 Disruption in mavericks** Disruption was computed using the formula in [5], which has the terms  $n_i$ ,  $n_j$ , and  $n_k$ . Each maverick experiment has nine mavericks (three of each type) and is characterized by a fixed out-degree for the mavericks (indicated by “od” followed by the fixed outdegree) and a fixed fitness value (indicated by “f” followed by the fitness value). The average of the three replicates is shown in the table.

## **S4 Computing a SASCA-ReS network with 218 million nodes**

Starting with the *sj\_er* seed set, which has 479027 nodes, SASCA-ReS is able to sustain an annual growth rate of 12% for 54 year and reach a network of size 217,839,850 nodes in 23h 20m 57s given 128 cores and 450GB of RAM. This resulting network had 9,390,741,826 edges.

## S5 Additional Details for Materials and Methods

We have previously generated a recency table according to the protocol described in [6] and publicly available in [7]. From this recency table, a small modification was made in which a value of 1 was added to all row entries such that no row contained a count of 0, ensuring that no year has a citation probability of 0.

## References

- [1] Chacko, G., Park, M., Ramavarapu, V., Grama, A., Robles-Granda, P., Warnow, T.: An agent-based model of citation behavior. (Accepted) *Applied Network Science* (2025). A preprint version is available on arXiv at <https://doi.org/10.48550/arXiv.2503.06579>
- [2] Park, M., Lamy, J.A., Rodrigues, E.C., Ferreira, F.M., Vu-Le, T.-A., Warnow, T., Chacko, G.: Very Large Scale Simulations of Network Growth with the Scalable Agent-based Simulator for Citation Analysis with sampling (SASCA-s). in press, 14th International Conference on Complex Networks and their Applications (2025) (2025)
- [3] Park, M., Vu-Le, T.-A., Warnow, T., Chacko, G.: SASCA-ReS. <https://github.com/illinois-or-research-analytics/SASCA-ReS>. Github repository (2025)
- [4] Efraimidis, P.S., Spirakis, P.G.: Approximation schemes for scheduling and covering on unrelated machines. *Theoretical Computer Science* **359**(1-3), 400–417 (2006)
- [5] Wu, L., Wang, D., Evans, J.A.: Large teams develop and small teams disrupt science and technology. *Nature* **566**(7744), 378–382 (2019) <https://doi.org/10.1038/s41586-019-0941-9>
- [6] Park, M., Vu-Le, T.-A., Warnow, T., Chacko, G.: Supplementary Materials for Very Large Scale Simulations of Network Growth with the Scalable Agent-based Simulator for Citation Analysis with sampling (SASCA-s). <https://github.com/illinois-or-research-analytics/SASCA/blob/main/docs/suppl.pdf> (2025)
- [7] Park, M., Chacko, G., Warnow, T., Vu-Le, T.-A., Rodrigues, E.C., Ferreira, F.M., Lamy, J.A.: Data from Development and Evaluation of SASCA-s: Scalable Agent-based Simulator for Citation Analysis With simulation. [https://doi.org/10.13012/B2IDB-3926377\\_V1](https://doi.org/10.13012/B2IDB-3926377_V1)