# A Puff of Steem: Security Analysis of Decentralized Content Curation

Aggelos Kiayias[1,2], Benjamin Livshits[3,4], Andrés Monteoliva
Mosteiro[1,5], and Orfeas Stefanos Thyfronitis Litos[1]

[1] University of Edinburgh
[2] IOHK
[3] Imperial College of London
[4] Brave Software
[5] Clearmatics
akiayias@inf.ed.ac.uk, ben@brave.com, amonteolivam@gmail.com,
o.thyfronitis@ed.ac.uk

**Abstract.** Decentralized content curation is the process through which
uploaded posts are ranked and filtered based exclusively on users' feed-
back. Platforms such as the blockchain-based Steemit[6] employ this type
of curation while providing monetary incentives to promote the visibil-
ity of high quality posts according to the perception of the participants.
Despite the wide adoption of the platform very little is known regarding
its performance and resilience characteristics. In this work, we provide
a formal model for decentralized content curation that identifies salient
complexity and game-theoretic measures of performance and resilience
to selfish participants. Armed with our model, we provide a first analy-
sis of Steemit identifying the conditions under which the system can be
expected to correctly converge to curation while we demonstrate its sus-
ceptibility to selfish participant behaviour. We validate our theoretical
results with system simulations in various scenarios.

## 1 Introduction

The modern Internet contains an immense amount of data; a single user
can only consume a tiny fraction in a reasonable amount of time. There-
fore, any widely used platform that hosts user-generated content (UGC)
must employ a content curation mechanism. Content curation can be un-
derstood as the set of mechanisms which rank, aggregate and filter rele-
vant information. In recent years, popular news aggregation sites like Red-
dit[7] or Hacker News[8] have established crowdsourced curation as the pri-
mary way to filter content for their users. Crowdsourced content curation,

---

[6] https://steemit.com/ Accessed: 2018-09-10

[7] https://www.reddit.com/ Accessed: 2018-09-25

[8] https://news.ycombinator.com/ Accessed: 2018-09-25

as opposed to more traditional techniques such as expert- or algorithmic-based curation, orders and filters content based on the ratings and feedback of the users themselves, obviating the need for a central moderator by leveraging the "wisdom of the crowd" [1].

The decentralized nature of crowdsourced curation makes it a suitable solution for ranking user-generated content in blockchain-based content hosting systems. The aggregation and filtering of user-generated content emerges as a particularly challenging problem in permissionless blockchains, as any solution that requires a concrete moderator implies that there exists a privileged party, which is incompatible with a permissionless blockchain. Moreover, public blockchains are easy targets for Sybil attacks, as any user can create new accounts at any time for a marginal cost. Therefore, on-chain mechanisms to resist the effect of Sybil users are necessary for a healthy and well-functioning platform; traditional counter-Sybil mechanisms [2] are much harder to apply in the case of blockchains due to the decentralized nature of the latter. The functions performed by moderators in traditional content platforms need to be replaced by incentive mechanisms that ensure self-regulation. Having the impact of a vote depend on the number of coins the voter holds is an intuitively appealing strategy to achieve a proper alignment of incentives for users in decentralized content platforms; specifically, it can render Sybil attacks impossible.

However, the correct design of such systems is still an unsolved problem. Blockchains have created a new economic paradigm where users are at the same time equity holders in the system, and leveraging this property in a robust manner constitutes an interesting challenge. A variety of projects have designed decentralized content curation systems [3,4,5]. Nevertheless, a deep understanding of the properties of such systems is still lacking. Among them, Steemit has a long track record, having been in operation since 2016 and attaining a user base of more than 1.08 M[9] registered accounts[10]. Steemit is a social media platform which lets users earn money (in the form of the STEEM cryptocurrency) by both creating and curating content in the network. Steemit is the front-end of the social network, a graphical web interface which allows users to see the content of the platform. On the other hand, all the back-end information is stored on a distributed ledger, the Steem blockchain. Steem can be understood as

---

[9] https://steemdb.com/accounts Accessed: 2018-09-25

[10] The number of accounts should not be understood as the number of active users, as one user can create multiple accounts.

an "app-chain", a blockchain with a specific application purpose: serving as a distributed database for social media applications [4].

In this work we initiate a study of the foundations of decentralized content curation. We develop a theoretical model of a post-voting system which aims to sort the posts created by users in a distributed and crowdsourced manner. Our model is constituted by a functionality which executes a protocol performed by $N$ players who have a fixed strategy, but in principle their set of actions could be extended. The $N$ players contribute votes in a round-based curation process. The impact of each vote depends on the number of coins held by the player. The posts are arranged in a list, sorted by the value of votes received, resembling the front-page model of Reddit or Hacker News. In the model, players vote according to their subjective opinion on the quality of the posts and have a limited attention span. Following previous related work, we represent each player's opinion on each post (i.e. likability) with a numerical value $l \in [0, 1]$ [6,1]. The objective quality of a post will be calculated as the simple summation of all players' likabilities for the post in question. To measure the properties of such a post-voting system, we say that the system $t$-converges if the $t$ first articles are ordered according to the objective quality of the posts at the end of the execution. After defining and formalizing an abstract post-voting system, we particularize it to model Steemit. We prove the conditions under which Steemit $t$-converges and then we develop a simulation to help us parametrize Steemit according to our post-voting model. Our analysis of the post-voting system of Steemit aims to provide a useful framework to aid the design of future decentralized curation platforms. [**Aggelos: This needs more work**]

## 2 Related Work

User-generated content (UGC) has been identified as a fundamental component of social media platforms and Web 2.0 in general [7]. The content created by users needs to be curated, and crowdsourced content curation [1] has emerged as an alternative to expert-based [8] or algorithmic-based [9] curation techniques. Motivated by the widespread adoption of crowdsourced aggregation sites such as Reddit or Digg[11], several research efforts [10,6,11] have aimed to model the mechanics and incentives for users in UGC platforms. This surge of interest is accompanied by empirical studies which have shown how social media users behave strategically when they publish and consume content [12]. As an example, in the case

---

[11] http://digg.com/ Accessed: 2018-09-25

of Reddit, users try to maximize their 'karma' [13], the social badge of the social media platform [14].

Previous works have analyzed content curation from an incentives and game-theoretic standpoint [6,10,15,12,11]. Our formalisation is based on these models and inherits features such as the quality distribution of the articles and the users' attention span [1,6]. In terms of measuring our results, the analysis of our *t-convergence* metric is similar to the top-$k$ posts in [1]. We also leverage the rank correlation coefficients Kendall's Tau [16] and Spearman's Rho [17] to measure content curation efficiency. Our approach describes the mechanics of post-voting systems from a computational perspective drawing inspiration from the real-ideal world paradigm of Cryptography [18,19] in the definition of $t$-convergence. Post-voting systems constitute a special case of voting mechanisms, as studied within social choice theory, belonging to the subcategory of cardinal voting systems [20]. We are aware of Gibbard's theorem [21]. To evade its complications, we constrain players to follow specific prescribed strategies. Our system shares the property of spanning multiple voting rounds with previous work [22]. Related literature [23,24,25] is centered on political elections and as a result attempts to resolve a variation of the problem with different constraints and assumptions. In the case of political elections, voter communication in many rounds is costly but navigating the ballot paper is free. In particular, voters can express their preferences for any candidate, irrespective of the order in which the latter appear on the ballot paper. On the other hand, the on-line and interactive nature of post-voting systems make multi-round voting sensible and even mandatory. We can thus consider a large number of rounds in the present work. The limitation in our case is that the average user is aware only of the most popular content; therefore she is unable to vote for posts that she has not viewed, e.g. because they are placed in the bottom of the list. We model this fact by using the concept of attention span, as detailed later.

Content curation is also related to the concept of online governance. The governance of online communities such as Wikipedia has been thoroughly studied in previous academic work [26,27]. However, the financially incentivized governance processes in blockchain systems, where the voters are at the same time equity-holders, have still many open research questions [28,29]. This shared ownership property has triggered interest in building social media platforms backed by distributed ledgers, where users are rewarded for generated content and variants of coin-holder voting are used to decide how these rewards are distributed. The effects of explicit financial incentives on the quality of content in Steemit has been

analyzed in [30]. Beyond the Steemit's whitepaper [4], a series of blog posts [31,32] effectively extend the economic analysis of the system. In parallel with Steemit, other projects such as Synereo[3] and Akasha[12] are exploring the convergence of social media and decentralized content curation. Beyond blockchain-based social media platforms, coin-holder voting systems are present in decentralized platforms such as DAOs [33] and in different blockchain protocols [34,35]. However, most of these systems use coin-holder voting processes to agree on a value or take a consensual decision.

## 3 Model

We first introduce some useful notation:

- We denote an ordered list of elements with $A = [e_1, \ldots, e_n]$ and the $i$-th element of the list with $A[i] = e_i$.
- Let $n \in \mathbb{N}^*$. $[n]$ denotes $\{1, 2, \ldots, n\}$.

### 3.1 Post list

**Definition 1 (Post).** *Let $N \in \mathbb{N}^*$. A post is defined as $p = (m, l)$, with $m \in [N], l \in [0,1]^N$.*

- **Author.** *The first element of a post is the id of its creator, $m$.*
- **Likability.** *The likability of a post is defined as $l \in [0,1]^N$.*

$N$ represents the number of voters (a.k.a players). A post has a distinct likability in $[0,1]$ for each player.

**Definition 2 (Ideal Score of a post).** *Let post $p = (m, l)$. We define the ideal score of $p$ as* $\text{idealSc}(p) = \sum_{i=1}^{|l|} l_i$.

The ideal score of a post is a single number that represents its overall worth to the community. By using simple summation, we assume that the opinions of all players have the same weight.

**Definition 3 (Post List).** *Let $M \in \mathbb{N}^*$. A post list $\mathcal{P} = [p_1, \ldots, p_M]$ is an ordered list containing posts. It may be the case that two posts are identical.*

---

[12] https://akasha.world/ Accessed: 2018-09-25

In the case of many UGC platforms, e.g. Steemit, there exists a feed (commonly named "Trending") that displays the same ordered posts for all users. In such an ordered list, posts placed closer to the top are more visible, since typical users consume content from top to bottom [**Orfeas: citation needed here?**] . We can thus measure the quality of an ordered list of posts by comparing it with a list that contains the same posts in decreasing order of ideal score.

**Definition 4 ($t$-Ideal Post Order).** *Let $\mathcal{P}$ a list of posts, $t \in \mathbb{N}^*$. We say that $\mathcal{P}$ is in $t$-ideal order and that the property $\text{IDEAL}^t(\mathcal{P})$ holds if*

$$\forall i < j \in t, \text{idealSc}(\mathcal{P}[i]) \geq \text{idealSc}(\mathcal{P}[j]) \quad .$$

*$\mathcal{P}$ is in $t$-ideal order iff $\text{IDEAL}^t(\mathcal{P})$ holds but $\text{IDEAL}^{t+1}(\mathcal{P})$ does not.*

## 3.2 Post Voting System

[**Orfeas: Next three paragraphs probably need changes**] We now define an abstract post-voting system. Such a system is defined through two Interactive Turing Machines (ITMs), $\mathcal{G}_{\text{Feed}}$ and $\Pi_{\text{honest}}$. The first controls the list of posts and aggregates votes, whereas one copy of the second ITM is instantiated for each player. $\mathcal{G}_{\text{Feed}}$ sends the post list to one player at a time, receives her vote and reorders the post list accordingly. The process is possibly repeated for many rounds.

A measure of the quality of a post-voting system is the maximum $t$ such that the post list at the end of the process is in $t$-ideal order.

In a more general setting, some of the honest protocol instantiations may be replaced with an arbitrary ITM. A robust post-voting system should still produce a post list of high quality.

**Definition 5 (Post-Voting System).** *Consider four PPT algorithms* INIT, AUX, HANDLEVOTE *and* VOTE*. The tuple $\mathcal{S}$ consisting of the four algorithms is a Post-Voting System. $\mathcal{S}$ parametrizes the following two ITMs:*

*$\mathcal{G}_{\text{Feed}}$ is a global functionality that accepts two messages:* `read`*, which responds with the current list of posts and* `vote`*, which can take various arguments and does whatever is defined in* HANDLEVOTE*.*

*$\Pi_{\text{honest}}$ is a protocol that sends* `read` *and* `vote` *messages to $\mathcal{G}_{\text{Feed}}$ whenever it receives (`activate`) from $\mathcal{E}$.*

**Algorithm 1** $\mathcal{G}_{\text{Feed}}\left(\text{INIT}, \text{AUX}, \text{HANDLEVOTE}\right)\left(\mathcal{P}, \text{initArgs}\right)$

1: Initialization:
2:      $\mathcal{U} \leftarrow \emptyset$                                            ▷ Set of players
3:      INIT (initArgs)

4:
5: Upon receiving (**read**) from $u_{\text{pid}}$:
6:      $\mathcal{U} \leftarrow \mathcal{U} \cup \{u_{\text{pid}}\}$
7:      aux $\leftarrow$ AUX $(u_{\text{pid}})$
8:      Send (**posts**, $\mathcal{P}$, aux) to $u_{\text{pid}}$

9:
10: Upon receiving (**vote**, ballot) from $u_{\text{pid}}$:
11:      HANDLEVOTE(ballot)

---

**Algorithm 2** $\Pi_{\text{honest}}\left(\text{VOTE}\right)$

1: Upon receiving (**activate**) from $\mathcal{E}$:
2:      Send (**read**) to $\mathcal{G}_{\text{Feed}}$
3:      Wait for response (**posts**, $\mathcal{P}$, aux)
4:      ballot $\leftarrow$ VOTE $(\mathcal{P}, \text{aux})$
5:      Send (**vote**, ballot) to $\mathcal{G}_{\text{Feed}}$

---

Players are activated by an Environment ITM that sends activation messages (Algorithm 2, line 1).

**Definition 6 (Post-Voting System Activation Message).** *We define* $\text{act}_{\text{pid}}$ *as the message* (**activate**, pid)*, sent to* $u_{\text{pid}}$*.*

**Definition 7 (Execution Pattern).** *Let* $N, R \in \mathbb{N}^*, N \geq 2$*.*

$$\text{ExecPat}_{N,R} = \big\{ (\text{act}_{\text{pid}_1}, \ldots, \text{act}_{\text{pid}_{NR}}) :$$
$$\forall i \in [R], \forall k \in [N], \exists j \in [N] : \text{pid}_{(i-1)N+j} = k \big\} \ ,$$

*i.e. activation messages are grouped in* $R$ *rounds and within each round each player is* **activate***d exactly once. The order of activations is not fixed.*

    *Let Environment* $\mathcal{E}$ *that sends messages* $\text{msgs} = (\text{act}_{\text{pid}_1}, \ldots, \text{act}_{\text{pid}_n})$ *sequentially. We say that* $\mathcal{E}$ *respects* $\text{ExecPat}_{N,R}$ *if* $\text{msgs} \in \text{ExecPat}_{N,R}$*. (Note: this implies that* $n = NR$*.)*

**Definition 8 (**$(N, R, M, t)$**-convergence under honesty).** *We say that a post-voting system* $\mathcal{S} = (\text{INIT}, \text{AUX}, \text{HANDLEVOTE}, \text{VOTE})$ $(N, R, M, t)$*-converges under honesty (or* $t$*-converges under honesty for* $N$ *players,* $R$

rounds and $M$ posts*) if, for every input $\mathcal{P}$ such that $|\mathcal{P}| = M$, for every $\mathcal{E}$ that respects $\text{ExecPat}_{N,R}$ and given that all protocols execute $\Pi_{\text{honest}}$, it holds that after $\mathcal{E}$ completes its execution pattern, $\mathcal{G}_{\text{Feed}}$ contains a post list $\mathcal{P}'$ such that $\text{IDEAL}^t(\mathcal{P}')$ is true.*

Note that concrete post voting systems may or may not give information such as the total number of rounds $R$ to the players. This is decided in algorithm Aux.

We now give a high-level description of a concrete post voting system, based on the Steemit platform. According to this mechanism, each player is assigned a number called "Voting Power" (VP) in $[0, 1]$, initialized to 1. A vote is a pair containing a post and a weight $w \in [0, 1]$. Upon receiving a list of posts, the honest player chooses to vote her most liked post amongst the top attSpan posts of the list. The weight is chosen to be equal to the respective likability. The functionality increases the score of the post by $a\text{VP}w + b$ and subsequently decreases the player's voting power by the same amount (but keeping it within the aforementioned bounds).

**Definition 9 (Steemit system).** *The Steemit system is the post voting system $\mathcal{S}$ with parameters $a, b, \text{regen} \in [0, 1] : a + b < 1, \left\lceil \frac{a+b}{\text{regen}} \right\rceil > 1, \text{attSpan} \in \mathbb{N}^*$. The four parametrizing procedures can be found in Appendix B.*

*Remark 1.* The constraint $a + b < 1$ ensures that a single vote of full weight cast by a player with full voting power does not completely deplete her voting power. The constraint $\left\lceil \frac{a+b}{\text{regen}} \right\rceil > 1$ excludes the degenerate case in which the regeneration of a single round is enough to fully replenish the voting power in all cases; in this case the purpose of voting power would be defeated.

*Remark 2.* The Steem blockchain protocol defines $a = 0.02, b = 0.0001$ and regen $= \frac{3}{5 \cdot 24 \cdot 60 \cdot 60} = 0.0000069\overline{4}$, thus $\left\lceil \frac{a+b}{\text{regen}} \right\rceil = 2895$.

*Remark 3.* We omit Steem Power, the voting-related currency of Steem, from our analysis.

*Remark 4.* Note (Algorithm 6, lines 24-40) that an honest player attempts to vote for as many posts as possible and spreads her votes with the maximum distance between them. The purpose of this is to efficiently utilize the available Voting Power to "make her voice heard". Also, efficiently using Voting Power on the Steemit website increases the voter's curation revenue [31].

**Theorem 1.** *The Steem system $(N, R, M, M)$-converges if and only if **SP** is constant and $R - 1 \geq (M - 1) \left\lceil \frac{a+b}{\text{regen}} \right\rceil$.*

*Proof Sketch.* When $R - 1 \geq (M - 1) \left\lceil \frac{a+b}{\text{regen}} \right\rceil$, there are enough rounds to ensure full regeneration of every player's voting power between two votes and thus the resulting post list reflects the true preferences of the players. In the opposite case, we can always craft a post list that exploits the fact that some votes are cast with reduced voting power in order to trick the system into placing a wrong post in the top position. $\square$

The above result is tight. If the conditions are violated the above theorem is not true. See Appendix A for full proof.

## 4 Simulation

### 4.1 Introduction

The previous outcomes are here complemented with experiments that verify our findings. We have implemented a simulation framework that realizes the execution of Steemit's post-voting system as defined above.

In particular, we consider two separate scenarios: First, we simulate the case when all players follow the prescribed honest strategy of Steemit, investigating how the curation quality of the system varies with the number of voting rounds. We successfully reproduce the result of Theorem 1, which implies that the system converges to the ideal post order up to the last post/completely/perfectly converges when a sufficient number of voting rounds is permitted, but otherwise the resulting list of posts may fail to achieve even 1-convergence.

The second case measures how resilient is the curation quality of Steemit against dishonest agents. Since a creator is financially rewarded when her content is upvoted, she has incentive to promote her own posts. A combination of in-band methods (apart from striving to produce posts of higher quality) can help her to that end. Voting for one's own posts, refraining from voting posts created by others and obtaining Sybil [36] accounts that only vote for her posts are only an indicative subset. We thus examine the quality of the resulting list when certain users do not follow the honest protocol, but apply the aforementioned self-promoting methods. We observe that a small increase in the number of selfish players has a detrimental effect to the t-convergence metric of the post voting system. Furthermore, we measure the number of positions on the list that the selfish post gains with respect to the number of selfish players.

## 4.2 Methodology

We leverage three metrics to compare the curated list with the ideal list: Kendall's Tau [16], Spearman's Rho [17], and $t$-ideal rank.

**Rank correlation coefficients.** To quantify the similarity between two ordered list of posts we employ Spearman's Rho and Kendall's Tau, two of the most popular rank correlation coefficients [16]. When measuring rank correlation, these coefficients will measure the statistical significance between two lists, each producing a value in $[-1, 1]$. When the ranking of two lists is completely unrelated to each other, the rank correlation coefficient will be 0. If there exists a correlation, the coefficients vary between 1 meaning absolute correlation (i.e. the the lists are identical) and -1, meaning absolute inverse correlation (i.e. the lists are sorted in reversed order).

In addition to the $t$-ideal rank and the rank correlation coefficients used in the first scenario, in the case of dishonest participants we include a metric that measures the gains of the selfish players. In particular, the metric is defined as the difference between the real position of the "selfish" post after the execution and its ranking according to the ideal order. We are thus able to measure how advantageous is for users to behave selfishly. Furthermore, $t$-ideal rank informs us how this behavior affects the overall quality of curation of the platform.

## 4.3 Results

In all simulations, the likabilities of all "honest" posts have been drawn from the $[0, 1]$-uniform distribution.

**Scenario A** As already mentioned, the results closely follow Theorem 1. Figures 1 and 2 show the $t$-ideal rank and Kendall's Tau coefficient respectively when the number of rounds is enough for all votes to be cast with full voting power. In particular, the parameters used are $a = \frac{1}{50}, b = 10^{-4}, \text{regen} = \frac{3}{5 \cdot 24 \cdot 60 \cdot 60}, R = 200000, \text{attSpan} = 10, N = 270$ and $M = 70$. (Observe that $R - 1 > (M - 1) \left\lceil \frac{a+b}{\text{regen}} \right\rceil$.)

As we can see, both measures show that the real list converges rapidly to the ideal order at the very end of the execution; in the meanwhile, the quality of the list improves very slowly.

Figures 3 and 4 depict what happens when the rounds are not sufficient for all votes to be cast with full voting power. In particular, the
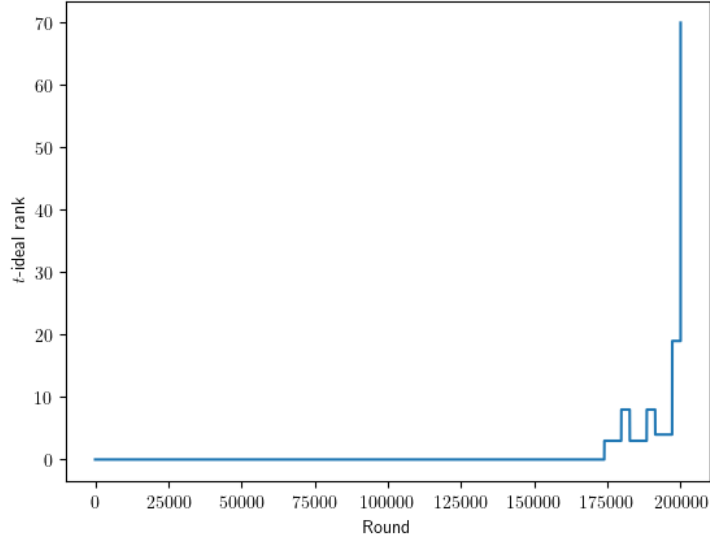
**Fig. 1.** $t$-ideal rank evolution with 270 honest players, 70 posts and 200.000 rounds
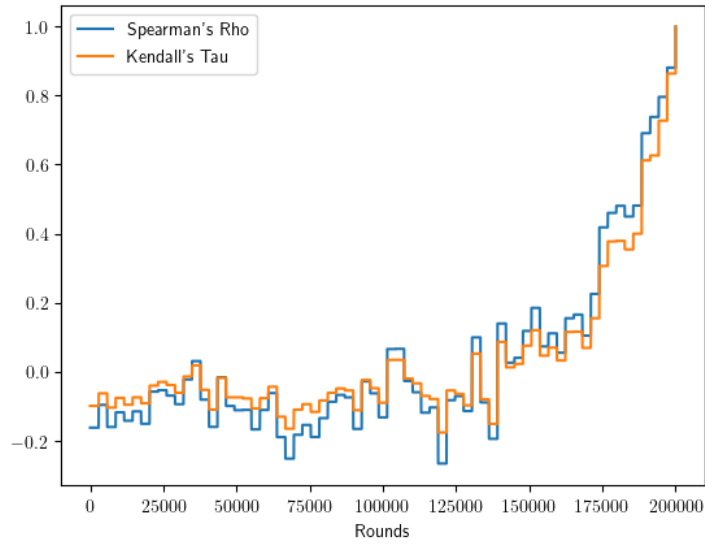


**Fig. 2.** Kendall's Tau and Spearman's Rho evolution with 270 honest players, 70 posts and 200.000 rounds

11

corresponding simulation was executed with the same parameters as before, only changing $M$ to 100 and $N$ to 300. (Observe that $R - 1 < (M - 1)\left\lceil\frac{a+b}{\text{regen}}\right\rceil$.)
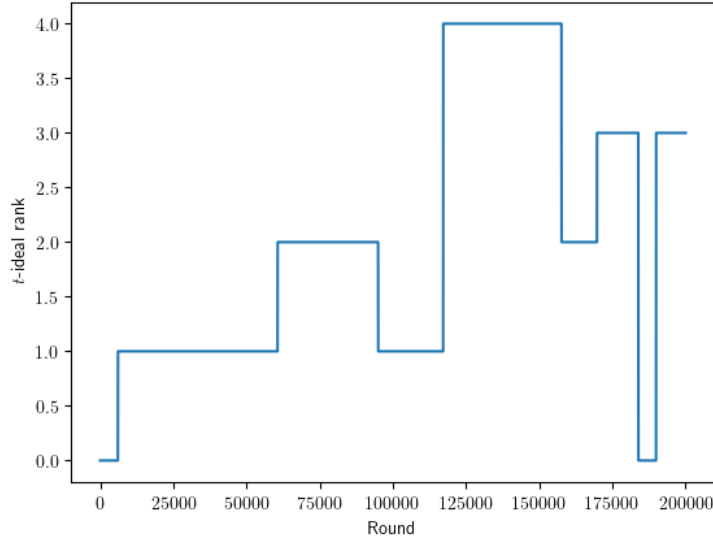


**Fig. 3.** $t$-ideal rank evolution with 300 honest players, 100 posts and 200.000 rounds

Here we see that at the end of the execution, only the first three posts are correctly ordered. Regarding the rest of the list, both Kendall's Tau and Spearman's Rho coefficients show that the order of the posts improves only slightly throughout the execution of the simulation.

**Scenario B: Selfish users.** In order to understand how the presence of voting rings/Sybil accounts affects the curation quality, we simulate the execution of the game for various voting ring sizes. We fix the rest of the system parameters so that the selfish post is handicapped. In particular, the voting rounds are sufficient for all votes to be cast with full voting power, the likability of the selfish post is 0 for all players and it is initially placed at the bottom of the post list. Define the gain of the post of the selfish players as its ideal position minus its final position. Figure 5 depicts the gain of the selfish post for a varying number of selfish players, from 1 to 100. Figure 6 shows the $t$-ideal rank of the resulting list at
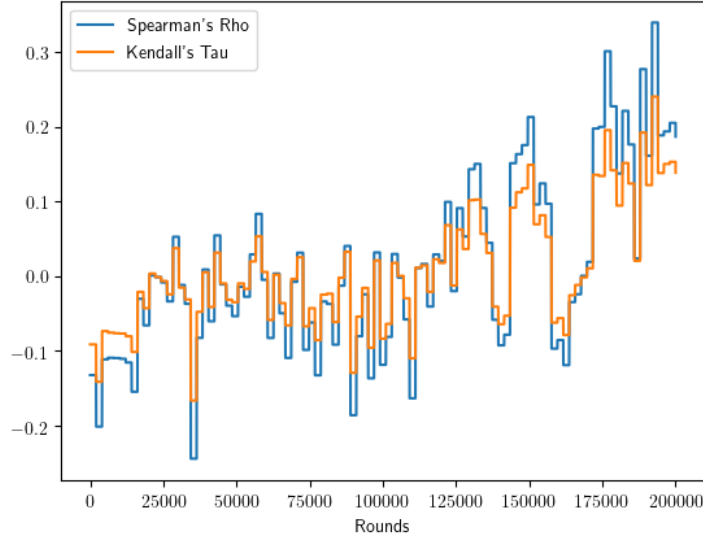
**Fig. 4.** Kendall's Tau and Spearman's Rho evolution with 300 honest players, 100 posts and 200.000 rounds

the same executions. For these simulations, the system parameters are $N = 101..200, a = \frac{1}{50}, b = 10^{-4}, \text{regen} = \frac{3}{5 \cdot 24 \cdot 60}, \text{attSpan} = 10, R = 5000$.

## 5 Results

We have defined an abstract post-voting system, along with a particularization inspired by the Steemit platform. We proved the exact conditions on the Steemit system parameters under which it successfully curates arbitrary lists of posts. We provided the results of simulations of the execution of the voting procedure under various conditions. Both cases with only honest and mixed honest and selfish players were simulated. We conclude that the Voting Power mechanism of Steem and the fact that self-voting is a profitable strategy may hurt curation quality.

## 6 Further Work

We have studied the curation properties of decentralized content curation platforms such as Steemit, obtaining new insights on the resilience of these systems. Some assumptions have been made in the presented
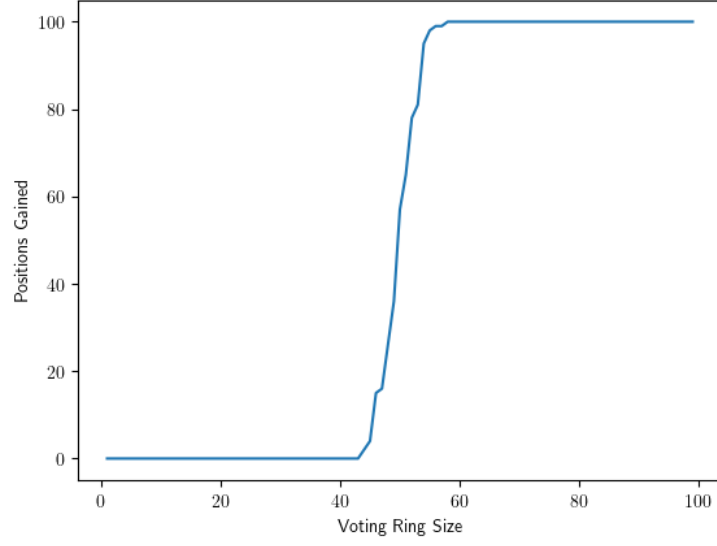
13

**Fig. 5.** Positions gained by selfish post with 100 honest players, 100 posts and 1 to 100 selfish players
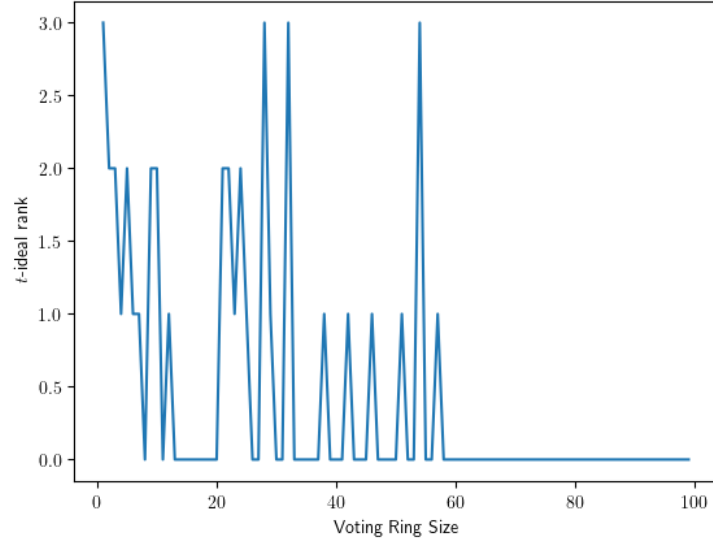


**Fig. 6.** $t$-ideal rank with 100 honest players, 100 posts and 1 to 100 selfish players

model. Various relaxations of these assumptions constitute fertile ground for future work. First of all, the selfish strategy can be extended and refined in various ways. For example, in a more general case voting rings can have the ability to create more than one posts in order to increase their rewards. Optimizing the number of posts and the vote allocation in this case would contribute towards a robust attack against the Steemit platform.

Selfish behavior is considered only in the simulation. Our theoretic results can be complemented with an analysis of games with selfish players.

Another open problem is the role of wealth distribution among players. In Steemit, the influence of the votes is weighed by the amount of coins each user has staked in the platform, a number known as Steem Power [4]. We have chosen to omit this concept, which effectively mitigates the possibility of Sybil attacks but increases the influence of wealthy individuals to the curation procedure. Including the distribution of wealth in our model may provide more realistic results and capture the relative influence of big stakeholders.

The addition of the economic factor invites the definition of utility functions and strategic behavior for the players. Its inclusion would imply the need for an expansion of our theorems and definitions to the strategic case, along with a full game-theoretic analysis. Furthermore, several possible refinements could be introduced; for example, the process of creating Sybil accounts could be associated with a monetary cost.

Last but not least, in our model posts are created only at the beginning of the execution. An extended model in which authors can create posts at any time and where the execution continues indefinitely (as is the case in real-world content curation systems) will provide further intuition on the topic of decentralized curation.

## 7  Acknowledgements

## Appendix A  Proof of Theorem 1: Steem convergence

*Proof.* − (⇐) Suppose that

$$R - 1 \geq (M - 1) \left\lceil \frac{a + b}{\text{regen}} \right\rceil \ . \tag{1}$$

Let pid $\in [N]$. In this case it is $R \geq M$ and according to VOTETHIS-ROUND in Algorithm 6, $u_\text{pid}$ votes non-null in rounds $(r_1, \ldots, r_M)$ with $r_i = \left\lfloor (i - 1) \frac{R-1}{M-1} \right\rfloor + 1$. Observe that:

$$(1) \Rightarrow \frac{R - 1}{M - 1} \geq \left\lceil \frac{a + b}{\text{regen}} \right\rceil \overset{\text{rhs}}{\underset{\text{integer}}{\Rightarrow}} \left\lfloor \frac{R - 1}{M - 1} \right\rfloor \geq \left\lceil \frac{a + b}{\text{regen}} \right\rceil \ , \tag{2}$$

$$\forall i \in [M] \setminus \{1\}, r_i \in \left\{ r_{i-1} + \left\lfloor \frac{R - 1}{M - 1} \right\rfloor, r_{i-1} + \left\lceil \frac{R - 1}{M - 1} \right\rceil \right\} \ . \tag{3}$$

From (2) and (3) we have that $\forall i \in [M - 1], r_{i+1} - r_i \geq \left\lceil \frac{a+b}{\text{regen}} \right\rceil$. We will now prove by induction that $\forall i \in [M], \mathbf{VP}_{\text{pid}, r_i} = 1$.

- For $i = 1, \mathbf{VP}_{\text{pid}, 1} = 1$ (Algorithm 3, line 4).
- Let $\mathbf{VP}_{\text{pid}, r_i} = 1$. Until $r_{i+1}$, a single non-null vote is cast by $u_\text{pid}$, which reduces $\mathbf{VP}_\text{pid}$ by at most $a + b$ (Algorithm 5, line 7) and at least $\left\lceil \frac{a+b}{\text{regen}} \right\rceil$ regenerations, each of which replenishes $\mathbf{VP}_\text{pid}$ by regen. Thus

$$\mathbf{VP}_{\text{pid}, r_{i+1}} \geq \min \left\{ \mathbf{VP}_{\text{pid}, r_i} - a - b + \text{regen} \left\lceil \frac{a + b}{\text{regen}} \right\rceil, 1 \right\} \geq 1 \ .$$

But $\mathbf{VP}_\text{pid}$ cannot exceed 1 (line 4), thus $\mathbf{VP}_{\text{pid}, r_{i+1}} = 1$.

Since the above holds for every pid $\in [N]$, it holds that at the end of the execution, all votes have been cast with full voting power, thus $\forall i \in [M], \text{sc}_R (\mathcal{P} [i]) = Nb + a \sum_{\text{pid}=1}^{N} \mathcal{P} [i]_\text{pid}$ and the posts in $\mathcal{P}_R$ are sorted by decreasing score (Algorithm 5, line 20). We observe that

$$\forall i \neq j \in [M], \text{idealSc} (\mathcal{P} [i]) > \text{idealSc} (\mathcal{P} [j]) \Rightarrow$$

$$\sum_{\text{pid}=1}^{N} \mathcal{P} [i]_\text{pid} > \sum_{\text{pid}=1}^{N} \mathcal{P} [j]_\text{pid} \Rightarrow$$

$$Nb + a \sum_{\text{pid}=1}^{N} \mathcal{P} [i]_\text{pid} > Nb + a \sum_{\text{pid}=1}^{N} \mathcal{P} [j]_\text{pid} \ .$$

Thus all posts will be ordered according to their ideal scores; put otherwise, IDEALSCORE$^M (\mathcal{P}_R)$ holds.

– ($\Rightarrow$) Suppose that

$$R - 1 < (M - 1) \left\lceil \frac{a + b}{\text{regen}} \right\rceil \ . \tag{4}$$

Several lists of posts will be defined in the rest of the proof. Given that, when all players are honest, the creator of a post is irrelevant, we omit the creator from the definition of posts to facilitate the exposition. Thus every post will be defined as a tuple of likabilities.

First, we consider the case when

$$\text{attSpan} + R \leq M \ . \tag{5}$$

In this case, no player can ever vote for the last post, as we will show now. First of all, $(5) \Rightarrow R < M$, thus all players cast $R$ votes in total. Let pid $\in N, i \in [R]$ and $v_{\text{pid},i}$ the index of the last post that has ever been in $u_{\text{pid}}$'s attention span until the end of round $i$, according to the ordering of $\mathcal{P}$. It is $v_{\text{pid},1} = \text{attSpan}$ and $\forall i \in [R] \setminus \{1\}, v_{\text{pid},i} = v_{\text{pid},i-1} + 1$, since in every round $u_{\text{pid}}$ votes for a single post and the first unvoted post of the list is added to their attention span. Note that, since this mechanism is the same for all players, the same unvoted post is added to all players' attention span at every round. Thus $\forall \text{pid} \in N, v_{\text{pid},R} = \text{attSpan} + R - 1 \overset{(5)}{<} M$. We deduce that no player has ever the chance to vote for the last post.

The above observation naturally leads us to the following counterexample: Let

$$\text{strongPost} = \Big( \underbrace{1, \ldots, 1}_{N} \Big)$$

$$\text{nullPost} = \Big( \underbrace{0, \ldots, 0}_{N} \Big)$$

$$\mathcal{P} = \Big[ \underbrace{\text{nullPost}, \ldots, \text{nullPost}}_{M-1}, \text{strongPost} \Big]$$

$\forall i \in [M-1]$, it is $\text{idealSc} \left( \mathcal{P}\left[M\right] \right) > \text{idealSc} \left( \mathcal{P}\left[M\right] \right)$, thus $\forall \mathcal{P}'$ that contain the same posts as $\mathcal{P}$ and $\text{IDEAL}^1 \left( \mathcal{P}' \right)$ holds, it is $\mathcal{P}'\left[1\right] = \mathcal{P}\left[M\right]$. However, since the last post is not voted by any player and the first post is voted by at least one player, it is $\text{sc}_R \left( \mathcal{P}\left[1\right] \right) > \text{sc}_R \left( \mathcal{P}\left[M\right] \right)$, thus $\text{IDEAL}^1 \left( \mathcal{P}_R \right)$ does not hold.

We now move on to the case when attSpan $+ R > M$. Let $V = \min\{R, M\}$. Each player casts exactly $V$ votes. Consider $\mathcal{P}^1 = 1^{M \times N}$ and pid $\in [N]$. Let

$$i \in [V] : \left( \mathbf{VPreg}_{\mathrm{pid},r_i} < 1 \wedge \nexists i' < i : \mathbf{VPreg}_{\mathrm{pid},r_{i'}} < 1 \right) \ ,$$

i.e. $i$ is the first round in which $u_{\mathrm{pid}}$ votes with less than full voting power. Such a round exists in every case as we will show now. Note that, since the first round is a voting round and the voting power of all players is full at the beginning, if $i$ exists it is is $i \geq 2$.

- If $R \geq M$, it is $V = M$.
  If $\nexists i \in [M] : \left( \mathbf{VPreg}_{\mathrm{pid},r_i} < 1 \wedge \nexists i' < i : \mathbf{VPreg}_{\mathrm{pid},r_{i'}} < 1 \right)$, then
  $\forall i \in [M], \mathbf{VPreg}_{\mathrm{pid},r_i} = 1 \Rightarrow \forall i \in [M] \setminus \{1\}, r_i \geq r_{i-1} + \left\lceil \frac{a+b}{\mathrm{regen}} \right\rceil$
  to have enough rounds to replenish the voting power after a full-weight, full-voting power vote. Thus $r_M \geq 1 + (M-1) \left\lceil \frac{a+b}{\mathrm{regen}} \right\rceil > R$, contradiction.

- If $R < M$, every player votes on all rounds, thus $r_2 = 2$. Note that

$$\left\lceil \frac{a+b}{\mathrm{regen}} \right\rceil \geq 2 \Rightarrow \frac{a+b}{\mathrm{regen}} > 1 \Rightarrow a + b > \mathrm{regen} \ . \tag{6}$$

Thus $\forall \mathrm{pid} \in [N], \mathbf{VPreg}_{\mathrm{pid},r_2} = 1 - a - b + \mathrm{regen} \overset{(6)}{<} 0$, thus $i = 2$. We proved that $i$ exists. Since all players follow the same voting pattern, the voting power of all players in each round is the same. Let $\mathrm{rVP} = \mathbf{VPreg}_{1,r_i}$. Assume that attSpan $< i \vee i > 2$. We cover the case where attSpan $\geq i \wedge i = 2$ later. In case $N$ is even, let $0 < \gamma < 0, 0 < \epsilon < \gamma(1 - \mathrm{rVP})$,

$$\mathrm{weakPost} = \left( \underbrace{1, \ldots, 1}_{N/2}, \underbrace{\gamma - \epsilon, \ldots, \gamma - \epsilon}_{N/2} \right) \ ,$$

$$\mathrm{strongPost} = \left( \underbrace{\gamma, \ldots, \gamma}_{N/2}, \underbrace{1, \ldots, 1}_{N/2} \right) \ ,$$

$$\mathrm{nullPost} = \left( \underbrace{0, \ldots, 0}_{N} \right) \ ,$$

$$\mathcal{P} = \left[ \underbrace{\mathrm{weakPost}, \ldots, \mathrm{weakPost}}_{i-1}, \mathrm{strongPost}, \underbrace{\mathrm{nullPost}, \ldots, \mathrm{nullPost}}_{M-i} \right] \ .$$

First of all, it is

$$\forall j \in [i-1], \mathrm{idealSc}\left(\mathcal{P}\left[j\right]\right) = \frac{N}{2}\left(1 + \gamma - \epsilon\right) <$$
$$< \frac{N}{2}\left(1 + \gamma\right) = \mathrm{idealSc}\left(\mathcal{P}\left[i\right]\right)$$

and $\forall j \in \{i+1, \ldots, M\}, \mathrm{idealSc}\left(\mathcal{P}\left[j\right]\right) = 0 < \mathrm{idealSc}\left(\mathcal{P}\left[i\right]\right)$, thus the strong post has strictly the highest ideal score of all posts and as a result, $\forall \mathcal{P}'$ that contains the same posts as $\mathcal{P}$ and $\mathrm{IDEAL}^1\left(\mathcal{P}'\right)$ holds, it is $\mathcal{P}'\left[1\right] = \mathcal{P}\left[i\right]$.

We observe that all players like both weak and strong posts more than null posts, thus no player will vote for a null post unless her attention span contains only null posts. This can happen in two cases: First, if the player has not yet voted for all non-null posts, but the first attSpan posts of the list, excluding already voted posts, are null posts. Second, if the player has already voted for all non-null posts. For a null post to rank higher than a non-null one, it must be true that there exists one player that has cast the first vote for the null post. However, since the null posts are initially at the bottom of the list and it is impossible for a post to improve its ranking before it is voted, we deduce that this first vote can be cast only after the voter has voted for all non-null posts. We deduce that all players vote for all non-null posts before voting for any null post.

We will now see that the first $\frac{N}{2}$ players vote first for all weak posts and then for the strong post. These players like the weak posts more than the strong post. As we saw, they will not vote any null post before voting for all non-null ones. If attSpan $> 1$ they vote for the strong post only when all other posts in their attention span are null ones and thus they will have voted for all weak posts already. If attSpan $= 1$ and since no post can increase its position before being voted, the strong post will become "visible" for all players only once they have voted for all weak posts. Thus in both cases the first $\frac{N}{2}$ players vote for the strong post only after they have voted for all weak posts first. The two previous results combined prove that the first $\frac{N}{2}$ players vote for the strong post in round $r_i$ exactly. We also observe that these players have experienced the exact same voting power reduction and regeneration as in the case of $\mathcal{P}^1$ since they voted only for posts with likeability 1, thus in round $r_i$ their voting power after regeneration is exactly the same as in the case of $\mathcal{P}^1 : \forall \mathrm{pid} \in \left[\frac{N}{2}\right], \mathbf{VP}\mathrm{reg}_{\mathrm{pid}, r_i} = \mathrm{rVP}$.

We observe that the first $\frac{N}{2}$ players vote for all weak posts with full voting power. As for the last $\frac{N}{2}$ players, we observe that, if attSpan $<$ $i$, they all vote for the first weak post of the list in the first round, and thus with full voting power. If attSpan $\geq i$ and $i > 2$, they vote for the strong post in the first round and for the first weak post in $r_2$ with full voting power. Thus in all cases the last $\frac{N}{2}$ players vote for the first weak post with full voting power. Therefore, the score of the first weak post at the end of the execution is $\mathrm{sc}_R\left(\mathcal{P}\left[1\right]\right) = \frac{N}{2}\left(a + b\right) + \frac{N}{2}\left(\left(\gamma - \epsilon\right)a + b\right)$.

On the other hand, at the end of the execution the strong post has been voted by the first $\frac{N}{2}$ players with rVP voting power and by the last $\frac{N}{2}$ players with at most full voting power, thus its final score will be at most $\mathrm{sc}_R\left(\mathcal{P}\left[i\right]\right) \leq \frac{N}{2}\left(\mathrm{rVP} \cdot \gamma a + b\right) + \frac{N}{2}\left(a + b\right)$. It is

$$\epsilon < \gamma\left(1 - \mathrm{rVP}\right) \Rightarrow$$
$$\frac{N}{2}\left(\mathrm{rVP} \cdot \gamma a + b\right) + \frac{N}{2}\left(a + b\right) < \frac{N}{2}\left(a + b\right) + \frac{N}{2}\left(\left(\gamma - \epsilon\right)a + b\right) \Rightarrow$$
$$\mathrm{sc}_R\left(\mathcal{P}\left[i\right]\right) < \mathrm{sc}_R\left(\mathcal{P}\left[1\right]\right) \ .$$

Thus $\mathcal{P}_R\left[1\right] \neq \mathcal{P}\left[i\right]$ and $\mathrm{Ideal}^1\left(\mathcal{P}_R\right)$ does not hold.

As for the case when $N$ is odd, let $0 < \epsilon < \gamma\frac{N-3}{N-1}\left(1 - \mathrm{rVP}\right)$. In this case, we assume that the likeability of the first $i$ posts (weak and strong) for the last player is $\gamma$, whereas the likeability of the last $M - i$ posts (the null posts) is 0. This means that the last player votes first for the weak and strong posts and then for the null posts. The rest of the likabilities remain as in the case when $N$ is even. We observe that the ideal score of the strong post is still strictly higher than the rest. Furthermore, since the last player votes for the first weak post within the first $i$ voting rounds, her voting power at the time of this vote will be at least rVP. We thus have the following bounds for the scores:

$$\mathrm{sc}_R\left(\mathcal{P}\left[i\right]\right) \leq \frac{N-1}{2}\left(\mathrm{rVP} \cdot \gamma a + b\right) + \frac{N-1}{2}\left(a + b\right) + \gamma a + b \ ,$$
$$\mathrm{sc}_R\left(\mathcal{P}\left[1\right]\right) \geq \frac{N-1}{2}\left(a + b\right) + \frac{N-1}{2}\left(\left(\gamma - \epsilon\right)a + b\right) + \mathrm{rVP} \cdot \gamma a + b \ .$$

Given the bounds of $\epsilon$, it is $\mathrm{sc}_R\left(\mathcal{P}\left[i\right]\right) < \mathrm{sc}_R\left(\mathcal{P}\left[1\right]\right)$, thus $\mathrm{Ideal}^1\left(\mathcal{P}_R\right)$ does not hold.

We finally cover the previously untreated edge case where attSpan $\geq$ $i \wedge i = 2$. rVP is defined like before. We first consider the case when

$N$ is even and greater than 2: $\exists k \in \mathbb{N} \setminus \{0,1\} : N = 2k$. Let $0 < \gamma < 1, 0 < \epsilon < 2\gamma \frac{1-\text{rVP}}{(k-1)\text{rVP}}$,

$$\text{weakPost} = \left( \underbrace{1, \ldots, 1}_{k-1}, \underbrace{\gamma - \epsilon, \ldots, \gamma - \epsilon}_{k-1}, \gamma, \gamma \right) ,$$

$$\text{strongPost} = \left( \underbrace{\gamma, \ldots, \gamma}_{k-1}, \underbrace{1, \ldots, 1}_{k-1}, \gamma, \gamma \right) ,$$

$$\mathcal{P} = \left[ \text{weakPost}, \text{strongPost}, \underbrace{\text{nullPost}, \ldots, \text{nullPost}}_{M-2} \right] .$$

We first observe that

$$\forall j \in \{3, \ldots, M\}, \text{idealSc}\left(\mathcal{P}\left[j\right]\right) = 0 <$$
$$< \text{idealSc}\left(\mathcal{P}\left[1\right]\right) = k - 1 + (k-1)(\gamma - \epsilon) + 2\gamma =$$
$$= k - 1 + (k+1)\gamma - (k-1)\epsilon <$$
$$< k - 1 + (k+1)\gamma = \text{idealSc}\left(\mathcal{P}\left[2\right]\right) ,$$

thus the strong post has strictly the highest ideal score of all posts and as a result, $\forall \mathcal{P}'$ that contains the same posts as $\mathcal{P}$ and $\text{IDEAL}^1\left(\mathcal{P}'\right)$ holds, it is $\mathcal{P}'\left[1\right] = \mathcal{P}\left[2\right]$.

The first $k - 1$ and the last two players vote first for $\mathcal{P}\left[1\right]$ and then for $\mathcal{P}\left[2\right]$, whereas players $k, \ldots, 2k - 2$ vote first for $\mathcal{P}\left[2\right]$ and then for $\mathcal{P}\left[1\right]$, thus at the end of the execution,

$$\text{sc}_R\left(\mathcal{P}\left[1\right]\right) = (k-1)(a+b) + 2(\gamma a + b) + (k-1)((\gamma - \epsilon)\text{rVP}a + b) ,$$
$$\text{sc}_R\left(\mathcal{P}\left[2\right]\right) = (k-1)(a+b) + (k+1)(\gamma \text{rVP}a + b) .$$

Given the bound on $\epsilon$, it is $\text{sc}_R\left(\mathcal{P}\left[1\right]\right) > \text{sc}_R\left(\mathcal{P}\left[2\right]\right)$, thus $\text{Ideal}^1\left(\mathcal{P}_R\right)$ does not hold.

Second, we consider the case when $N$ is odd: $\exists k \in \mathbb{N} : N = 2k + 1$. Let $0 < \gamma < 1, 0 < \epsilon < \gamma \frac{1 - \mathrm{rVP}}{k \mathrm{rVP}}$,

$$\mathrm{weakPost} = \left( \underbrace{1, \ldots, 1}_{k}, \underbrace{\gamma - \epsilon, \ldots, \gamma - \epsilon}_{k}, \gamma \right) \quad,$$

$$\mathrm{strongPost} = \left( \underbrace{\gamma, \ldots, \gamma}_{k}, \underbrace{1, \ldots, 1}_{k}, \gamma \right) \quad,$$

$$\mathcal{P} = \left[ \mathrm{weakPost}, \mathrm{strongPost}, \underbrace{\mathrm{nullPost}, \ldots, \mathrm{nullPost}}_{M-2} \right] \quad.$$

We first observe that

$$\forall j \in \{3, \ldots, M\}, \mathrm{idealSc}\,(\mathcal{P}\,[j]) = 0 <$$
$$< \mathrm{idealSc}\,(\mathcal{P}\,[1]) = k + k\,(\gamma - \epsilon) + \gamma =$$
$$= k + (k+1)\,\gamma - k\epsilon < k + (k+1)\,\gamma = \mathrm{idealSc}\,(\mathcal{P}\,[2]) \quad,$$

thus the strong post has strictly the highest ideal score of all posts and as a result, $\forall \mathcal{P}'$ that contains the same posts as $\mathcal{P}$ and $\mathrm{IDEAL}^1\,(\mathcal{P}')$ holds, it is $\mathcal{P}'\,[1] = \mathcal{P}\,[2]$.

The first $k$ and the last player vote first for $\mathcal{P}\,[1]$ and then for $\mathcal{P}\,[2]$, whereas players $k+1, \ldots, 2k$ vote first for $\mathcal{P}\,[2]$ and then for $\mathcal{P}\,[1]$, thus at the end of the execution,

$$\mathrm{sc}_R\,(\mathcal{P}\,[1]) = k\,(a+b) + \gamma a + b + k\,((\gamma - \epsilon)\,\mathrm{rVP}a + b) \quad,$$
$$\mathrm{sc}_R\,(\mathcal{P}\,[2]) = k\,(a+b) + (k+1)\,(\gamma\mathrm{rVP}a + b) \quad.$$

Given the bound on $\epsilon$, it is $\mathrm{sc}_R\,(\mathcal{P}\,[1]) > \mathrm{sc}_R\,(\mathcal{P}\,[2])$, thus $\mathrm{Ideal}^1\,(\mathcal{P}_R)$ does not hold.

Last but not least, we consider the case when $N = 2$. In this case, let $0 < \gamma < 1$ and

$$\mathcal{P} = \left[ (1, 0), \left( \gamma, 1 - \gamma \frac{1 + \mathrm{rVP}}{2} \right), \underbrace{\mathrm{nullPost}, \ldots, \mathrm{nullPost}}_{M-2} \right] \quad.$$

It is $\forall j \in \{3, \ldots, M\}, \mathrm{idealSc}\,(\mathcal{P}\,[j]) = 0 < \mathrm{idealSc}\,(\mathcal{P}\,[1]) = 1 \overset{\mathrm{rVP} < 1}{<}$ $1 + \gamma \frac{1 - \mathrm{rVP}}{2} = \gamma + 1 - \gamma \frac{1 + \mathrm{rVP}}{2} = \mathrm{idealSc}\,(\mathcal{P}\,[2])$, thus $\mathcal{P}\,[2]$ has strictly

22

the highest ideal score of all posts and as a result, $\forall \mathcal{P}'$ that contains the same posts as $\mathcal{P}$ and $\text{IDEAL}^1(\mathcal{P}')$ holds, it is $\mathcal{P}'[1] = \mathcal{P}[2]$.

On the other hand, $\text{sc}_R(\mathcal{P}[1]) = a + 2b > \gamma \text{rVP} a + b + \left(1 - \gamma \frac{1+\text{rVP}}{2}\right) a + b = \text{sc}_R(\mathcal{P}[2])$, thus $\text{Ideal}^1(\mathcal{P}_R)$ does not hold.

$\square$

## Appendix B    Steem post voting system procedures

---
**Algorithm 3** $\text{INIT}(\text{attSpan}, a, b, \text{regen}, R)$

---
1: Store input parameters as constants
2: $r \leftarrow 1$
3: lastVoted $\leftarrow (0, \dots, 0) \in (\mathbb{N}^*)^N$
4: $\mathbf{VP} \leftarrow (1, \dots, 1) \in [0,1]^N$
5: scores $\leftarrow (0, \dots, 0) \in (\mathbb{R}^+)^M$

---

---
**Algorithm 4** $\text{AUX}$

---
1: **return** $(\text{attSpan}, a, b, r, \text{regen}, R)$

---

**Algorithm 5** HANDLEVOTE $(\text{ballot}, u_{\text{pid}})$

1: **if** $\text{lastVoted}_{\text{pid}} \neq r$ **then**                 ▷ One vote per player per round
2:      $\mathbf{VP}_{\text{pid},r} \leftarrow \mathbf{VP}_{\text{pid}}$                               ▷ For proofs
3:      $\mathbf{VP}_{\text{pid}} \leftarrow \max\{\mathbf{VP}_{\text{pid}} + \text{regen}, 1\}$
4:      $\mathbf{VP}\text{reg}_{\text{pid},r} \leftarrow \mathbf{VP}_{\text{pid}}$                        ▷ For proofs
5:      **if** $\text{ballot} \neq \textbf{null}$ **then**
6:          Parse ballot as $(p, \text{weight})$
7:          $\text{cost} \leftarrow a \cdot \mathbf{VP}_{\text{pid}} \cdot \text{weight} + b$
8:          **if** $\mathbf{VP}_{\text{pid}} - \text{cost} \geq 0$ **then**
9:              $\text{score} \leftarrow \text{cost} \cdot \mathbf{SP}_{\text{pid}}$
10:             $\mathbf{VP}_{\text{pid}} \leftarrow \mathbf{VP}_{\text{pid}} - \text{cost}$
11:          **else**
12:              $\text{score} \leftarrow \mathbf{VP}_{\text{pid}} \cdot \mathbf{SP}_{\text{pid}}$
13:             $\mathbf{VP}_{\text{pid}} \leftarrow 0$
14:          **end if**
15:          $\text{scores}_p \leftarrow \text{scores}_p + \text{score}$
16:      **end if**
17:      $\text{lastVoted}_{\text{pid}} \leftarrow r$
18: **end if**
19: **if** $\forall i \in [N], \text{lastVoted}_i = r$ **then**                  ▷ round over
20:      $\mathcal{P} \leftarrow \text{ORDER}(\mathcal{P}, \text{scores})$             ▷ order posts by votes
21:      $\mathcal{P}_r \leftarrow \mathcal{P}$                                  ▷ For proofs
22:      $r \leftarrow r + 1$
23: **end if**     ▷ TODO: count rounds? simplify with set of voted and check of length?

**Algorithm 6** VOTE $(\mathcal{P}, \mathrm{aux})$

---

1: Store aux contents as constants
2: voteRounds $\leftarrow$ VOTEROUNDS $(R, |\mathcal{P}|)$
3: **if** VOTETHISROUND $(r, |\mathcal{P}|) = $ yes **then**
4:      top $\leftarrow$ CHOOSETOPPOSTS $(\mathrm{attSpan}, \mathcal{P}, \mathrm{votedPosts})$
5:      $(i, l) \leftarrow \underset{(i,l) \in \mathrm{top}}{\mathrm{argmax}} \{l_{\mathrm{pid}}\}[1]$
6:      votedPosts $\leftarrow$ votedPosts $\cup (i, l)$
7:      **return** $((i, l), l_{\mathrm{pid}})$
8: **else**
9:      **return null**
10: **end if**
11:
12: **function** CHOOSETOPPOSTS(attSpan, $\mathcal{P}$, votedPosts)
13:      res $\leftarrow \emptyset$
14:      idx $\leftarrow 1$
15:      **while** $|\mathrm{res}| < \mathrm{attSpan}$ & idx $\leq |\mathcal{P}|$ **do**
16:          **if** $\mathcal{P}[\mathrm{idx}] \notin \mathrm{votedPosts}$ **then**              $\triangleright$ One vote per post per player
17:              res $\leftarrow$ res $\cup \{\mathcal{P}[\mathrm{idx}]\}$
18:          **end if**
19:          idx $\leftarrow$ idx $+ 1$
20:      **end while**
21:      **return** res
22: **end function**
23:
24: **function** VOTETHISROUND$(r, M)$
25:      **if** $R < M$ **then**
26:          **return** yes
27:      **else if** $r \in \mathrm{voteRounds}$ **then**
28:          **return** yes
29:      **else**
30:          **return** no
31:      **end if**
32: **end function**
33:
34: **function** VOTEROUNDS$(R, M)$
35:      voteRounds $\leftarrow \emptyset$
36:      **for** $i = 1$ to $M$ **do**
37:          voteRounds $\leftarrow$ voteRounds $\cup \left\{ 1 + \left\lfloor (i - 1) \frac{R-1}{M-1} \right\rfloor \right\}$
38:      **end for**
39:      **return** voteRounds
40: **end function**

---

# References

1. Askalidis G., Stoddard G.: A theoretical analysis of crowdsourced content curation. In The 3rd Workshop on Social Computing and User Generated Content: vol. 16 (2013)

2. Levine B. N., Shields C., Margolin N. B.: A survey of solutions to the sybil attack. University of Massachusetts Amherst, Amherst, MA: vol. 7, p. 224 (2006)

3. Konforty D., Adam Y., Estrada D., Meredith L. G.: Synereo: The Decentralized and Distributed Social Network. Self-published: `https://pdfs.semanticscholar.org/253c/c4744e6b2b87f88e46188fe527982b19542e.pdf` Accessed: 2018-08-04. (2015)

4. Steem Whitepaper. `https://steem.io/steem-whitepaper.pdf` Accessed: 2018-08-05. (2018)

5. Goldin M.: Token-Curated Registries 1.0. `https://medium.com/@ilovebagels/token-curated-registries-1-0-61a232f8dac7` Accessed: 2018-08-13. (2017)

6. Ghosh A., McAfee P.: Incentivizing high-quality user-generated content. In Proceedings of the 20th international conference on World wide web: pp. 137–146: ACM (2011)

7. Kaplan A. M., Haenlein M.: Users of the world, unite! The challenges and opportunities of Social Media. Business horizons: vol. 53(1), pp. 59–68 (2010)

8. Stanoevska-Slabeva K., Sacco V., Giardina M.: Content Curation: a new form of gatewatching for social media. Documento electrónico. Recuperado el: vol. 16 (2012)

9. Rader E., Gray R.: Understanding user beliefs about algorithmic curation in the Facebook news feed. In Proceedings of the 33rd annual ACM conference on human factors in computing systems: pp. 173–182: ACM (2015)

10. Das Sarma A., Das Sarma A., Gollapudi S., Panigrahy R.: Ranking mechanisms in twitter-like forums. In Proceedings of the third ACM international conference on Web search and data mining: pp. 21–30: ACM (2010)

11. Abbassi Z., Hegde N., Massoulié L.: Distributed content curation on the Web. ACM Transactions on Internet Technology (TOIT): vol. 14(2-3), p. 9 (2014)

12. May A., Chaintreau A., Korula N., Lattanzi S.: Filter & follow: How social media foster content curation. In ACM SIGMETRICS Performance Evaluation Review: vol. 42: pp. 43–55: ACM (2014)

13. Bergstrom K.: "Don't feed the troll": Shutting down debate about community expectations on Reddit. com. First Monday: vol. 16(8) (2011)

14. Anderson A., Huttenlocher D., Kleinberg J., Leskovec J.: Steering user behavior with badges. In Proceedings of the 22nd international conference on World Wide Web: pp. 95–106: ACM (2013)

15. Gupte M., Hajiaghayi M., Han L., Iftode L., Shankar P., Ursu R. M.: News posting by strategic users in a social network. In International Workshop on Internet and Network Economics: pp. 632–639: Springer (2009)

16. Kendall M. G.: Rank correlation methods (1955)

17. Spearman C.: The proof and measurement of association between two things. The American journal of psychology: vol. 15(1), pp. 72–101 (1904)

18. Goldreich O.: The foundations of modern cryptography. In Modern Cryptography, Probabilistic Proofs and Pseudorandomness: pp. 1–37: Springer (1999)

19. Lindell Y., Katz J.: Introduction to modern cryptography. Chapman and Hall/CRC (2014)

20. Hillinger C.: The case for utilitarian voting (2005)

21. Gibbard A.: Manipulation of voting schemes: a general result. Econometrica: journal of the Econometric Society: pp. 587–601 (1973)

22. Kalech M., Kraus S., Kaminka G. A., Goldman C. V.: Practical voting rules with partial information. Autonomous Agents and Multi-Agent Systems: vol. 22(1), pp. 151–182 (2011)

23. Lu T., Boutilier C.: Robust approximation and incremental elicitation in voting protocols. In IJCAI: vol. 1: pp. 287–293 (2011)
24. Conitzer V., Sandholm T.: Communication complexity of common voting rules. In Proceedings of the 6th ACM conference on Electronic commerce: pp. 78–87: ACM (2005)
25. Xia L., Conitzer V.: Compilation Complexity of Common Voting Rules. In AAAI (2010)
26. Leskovec J., Huttenlocher D. P., Kleinberg J. M.: Governance in social media: A case study of the wikipedia promotion process. In ICWSM: pp. 98–105 (2010)
27. Forte A., Bruckman A.: Scaling consensus: Increasing decentralization in Wikipedia governance. In Hawaii International Conference on System Sciences, Proceedings of the 41st Annual: pp. 157–157: IEEE (2008)
28. Buterin V.: Notes on Blockchain Governance. `https://vitalik.ca/general/2017/12/17/voting.html` Accessed: 2018-09-04. (2017)
29. Ehrsam F.: Blockchain Governance: Programming Our Future. `https://www.medium.com`: `https://www.medium.com/@FEhrsam/blockchain-governance-programming-our-future-c3bfe30f2d74` Accessed: 2018-09-03 (2017)
30. Thelwall M.: Can social news websites pay for content and curation? The SteemIt cryptocurrency model. Journal of Information Science: p. 0165551517748290 (2017)
31. González J.: Author and Curator rewards in HF19. `https://steemit.com/steemit/@jga/author-and-curator-rewards-in-hf19`. Accessed: 2018-07-01 (2018)
32. González J.: Self-voters can achieve an interest of 248% APR!! `https://steemit.com/utopian-io/@jga/self-voters-can-achieve-an-interest-of-248-apr`. Accessed: 2018-07-05 (2018)
33. Philip Daian Tyler Kell I. M., Juels A.: On-Chain Vote Buying and the Rise of Dark DAOs. `http://hackingdistributed.com/2018/07/02/on-chain-vote-buying/` Accessed: 2018-09-06. (2018)
34. Duffield E., Diaz D.: Dash: A PrivacyCentric CryptoCurrency. Self-published (2015)
35. Goodman L.: Tezos–a self-amending crypto-ledger White paper. `https://www.tezos.com/static/papers/white_paper.pdf` (2014)
36. Douceur J. R.: The Sybil Attack. International workshop on Peer-To-Peer Systems (2002)