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


## Aggelos Kiayias
University of Edinburgh, United Kingdom
IOHK, Hong Kong
akiayias@inf.ed.ac.uk

## Benjamin Livshits
Imperial College of London, United Kingdom
Brave Software, United Kingdom
ben@brave.com

## Andrés Monteoliva Mosteiro
University of Edinburgh, United Kingdom
Clearmatics, United Kingdom
amonteolivam@gmail.com

## Orfeas Stefanos Thyfronitis Litos
University of Edinburgh, United Kingdom
o.thyfronitis@ed.ac.uk



## Abstract

Decentralized content curation is the process through which uploaded posts are ranked and filtered based exclusively on users' feedback. Platforms such as the blockchain-based Steemit[1] employ this type of curation while providing monetary incentives to promote the visibility 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 analysis of Steemit identifying the conditions under which the system can be expected to correctly converge to curation while we demonstrate its susceptibility to selfish participant behaviour. We validate our theoretical results with system simulations in various scenarios.



**2012 ACM Subject Classification** Security and privacy → Distributed systems security

**Keywords and phrases** blockchain, content curation, decentralized, voting

**Digital Object Identifier** 10.4230/OASIcs...

**Related Version** A full version of the paper is available at https://arxiv.org/abs/1810.01719.


## 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. Therefore, any widely used platform that hosts user-generated content (UGC) must employ a content curation mechanism. Content curation can be understood as the set of mechanisms which rank, aggregate and filter relevant information. In recent years, popular news aggregation sites like Reddit[2] or Hacker News[3] have established crowdsourced curation as the primary way to filter content for their users. Crowdsourced content curation, as opposed to more traditional techniques such as expert- or

[1] https://steemit.com/ Accessed: 2019-01-02
[2] https://www.reddit.com/ Accessed: 2019-01-02
[3] https://news.ycombinator.com/ Accessed: 2019-01-02

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" [3].

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 [29] 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 [27, 39, 16]. 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[4] registered accounts[5]. 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 an "app-chain", a blockchain with a specific application purpose: serving as a distributed database for social media applications [39].

## 1.1 Our Contributions

In this work we study the foundations of decentralized content curation from a computational perspective. We develop an abstract 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. The model includes an honest participant behaviour while it allows deviations to be modeled for a subset of the participants. 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 [14, 3], we represent each player's opinion on each post (i.e. likability) with a numerical value $l \in [0, 1]$. The objective quality of a post is calculated

[4] https://steemdb.com/accounts Accessed: 2019-01-02
[5] The number of accounts should not be understood as the number of active users, as one user can create multiple accounts.

as the simple summation of all players' likabilities for the post in question. To measure the effectiveness of a post-voting system, we introduce the property of *convergence* under honesty which is parameterised by a number of values including a metric $t$, that demands the first $t$ articles to be ordered according to the objective quality of the posts at the end of the execution assuming all participants signal honestly to the system their personal preferences. Armed with our post-voting system abstraction, we proceed to particularize it to model Steemit and provide the following results.

i) We characterise the conditions under which the Steemit algorithm converges under honesty. Our results highlight some fundamental limitations of the actual Steemit parameterization. Specifically, for curated lists of length bigger than 70 the algorithm may *not achieve even 1-convergence*.

ii) We validate our results with a simulation testing different metrics based on correlation that have been proposed in previous works [25, 35] and relating them to our notion of convergence.

iii) We demonstrate that "selfish" deviation from honest behavior results to substantial gains in terms of boosting the ranking of specific posts in the resulting list of the post-voting system, and to a grave reduction of the quality of said list.

### 1.2 Steem consensus algorithm

In a nutshell, the Delegated Proof of Stake [8, 34, 38] works as follows: Steem users can sign up as "validator" candidates for one of 21 slots. Each user that owns some STEEM can vote for a validator. The 20 candidates that receive most votes (weighted by the respective users' STEEM) become validators. The 21st slot is filled with one of the candidates that was not elected, chosen at random with probability proportional to her votes.

A validator is responsible for receiving new transactions and adding them to blocks. Validators take turns in block production. An honest validator attaches her block to the latest valid block she knows and broadcasts it to the network. We say that a round is complete after each validator has had a chance to create a block. Honest nodes accept the longest known chain as the valid one. Elections for validators happen once each round, thus each STEEM holder is allowed to change her opinion very often.

The protocol promises that all new transactions are permanently added to the blockchain in a short amount of time, given that at least two thirds of the validators are honest. Unfortunately, we were unable to locate a formal proof of this claim.

Note that our analysis does not focus on DPOS, but on the curation mechanism of Steemit. The latter is independent of the consensus protocol of Steem.

## 2 Related Work

User-generated content (UGC) has been identified as a fundamental component of social media platforms and Web 2.0 in general [24]. The content created by users needs to be curated, and crowdsourced content curation [3] has emerged as an alternative to expert-based [36] or algorithmic-based [33] curation techniques. Motivated by the widespread adoption of crowdsourced aggregation sites such as Reddit or Digg[6], several research efforts [9, 14, 1] have aimed to model the mechanics and incentives for users in UGC platforms. This surge

[6] `http://digg.com/` Accessed: 2019-01-02

of interest is accompanied by studies which have shown how social media users behave strategically when they publish and consume content [32]. As an example, in the case of Reddit, users try to maximize their 'karma' [4], the social badge of the social media platform [2].

Previous works have analyzed content curation from an incentives and game-theoretic standpoint [14, 9, 21, 32, 1] . Our formalisation is based on these models and inherits features such as the quality distribution of the articles and the users' attention span [3, 14]. In terms of the analysis of our results, the analysis of our *t-convergence* metric is similar to the top-$k$ posts in [3]. We also leverage the rank correlation coefficients Kendall's Tau [25] and Spearman's Rho [35] to measure content curation efficiency. Our approach describes the mechanics of post-voting systems from a computational perspective, something that departs from the approach of all previous works, drawing inspiration from the real-ideal world paradigm of cryptography [17, 30] as employed in our 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 [22]. In this context, it follows from Gibbard's theorem [15] that no decentralised non-trivial post-voting mechanism can be strategy-proof. This is consistent with our results that demonstrate how selfish behaviour is beneficial to the participants. Our system shares the property of spanning multiple voting rounds with previous work [23]. Other related literature in social choice [31, 6, 40] is centered on political elections and as a result attempts to resolve a variation of the problem with quite different constraints and assumptions. In more detail, in the case of political elections, voter communication in many rounds is costly while navigating the ballot is not subject to any constraints as voters are assumed to have plenty of time to parse all the options available to them. As a result, 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 online and interactive nature of post-voting systems make multi-round voting a natural feature to be taken advantage of. At the same time, the fairness requirements are more lax and it is acceptable (even desirable) for participants to act reactively on the outcome of each others' evaluations. On the other hand, in the post-voting case, the "ballot" is only partially available given the high number of posts to be ranked that may very well exceed the time available to a (human) user to participate in the process. As a result a user will be unable to vote for posts that she has not viewed, for instance, because they are placed in the bottom of the list. This is captured in our model by introducing the concept of "attention span."

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 [28, 13]. 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 [5, 12]. 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 [37]. Beyond the Steemit's whitepaper [39], a series of blog posts [18, 19] effectively extend the economic analysis of the system. In parallel with Steemit, other projects such as Synereo [27] and Akasha[7] are exploring the convergence of social media and decentralized content curation. Beyond blockchain-based social media platforms, coin-holder

[7] `https://akasha.world/` Accessed: 2019-01-02

voting systems are present in decentralized platforms such as DAOs [7] and in different blockchain protocols [11, 20]. 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* $\mathrm{idealSc}(P) = \sum_{i=1}^{|l|} l_i$.

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.*

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 users typically consume content from top to bottom. 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, $t \in [M]$. The property* $\textsc{Ideal}^t(\mathcal{P})$ *holds if*

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

*We say that $\mathcal{P}$ has a $t$-*ideal rank *if* $\textsc{Ideal}^t(\mathcal{P})$ *holds and $t$ is the maximum integer less or equal to $M$ with this property.*

## 3.2 Post Voting System

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 $t$-ideal rank of the post list at the end of the process.

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* $\textsc{Init}, \textsc{Aux}, \textsc{HandleVote}$ *and* $\textsc{Vote}$. *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* $\textsc{HandleVote}$.

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

**Algorithm 1** $\mathcal{G}_{\text{Feed}}\left(\textsc{Init}, \textsc{Aux}, \textsc{HandleVote}\right)(\mathcal{P}, \text{initArgs})$

```
Initialization:
    𝒰 ← ∅                                        ▷ Set of players
    Init (initArgs)

Upon receiving (read) from u_pid:
    𝒰 ← 𝒰 ∪ {u_pid}
    aux ← Aux (u_pid)
    Send (posts, 𝒫, aux) to u_pid

Upon receiving (vote, ballot) from u_pid:
    HandleVote(ballot)
```

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

```
1: Upon receiving (activate) from ℰ:
2:     Send (read) to 𝒢_Feed
3:     Wait for response (posts, 𝒫, aux)
4:     ballot ← Vote (𝒫, aux)
5:     Send (vote, ballot) to 𝒢_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* $\texttt{act}_{\text{pid}}$ *as the message* $(\texttt{activate}, \text{pid})$, *sent to* $u_{\text{pid}}$.

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

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

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

*Let Environment $\mathcal{E}$ that sends messages* $\text{msgs} = \left(\texttt{act}_{\text{pid}_1}, \dots, \texttt{act}_{\text{pid}_n}\right)$ *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* $\mathcal{S} = (\textsc{Init}, \textsc{Aux}, \textsc{HandleVote}, \textsc{Vote})$ $(N, R, M, t)$-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* $\mathrm{ExecPat}_{N,R}$ *and given that all protocols execute* $\Pi_{\mathrm{honest}}$*, it holds that after* $\mathcal{E}$ *completes its execution pattern,* $\mathcal{G}_{\mathrm{Feed}}$ *contains a post list* $\mathcal{P}'$ *such that* $\textsc{Ideal}^t(\mathcal{P}')$ *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 $\textsc{Aux}$.

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 of coins known as "Steem Power" (SP) that remains constant throughout the execution and another number called "Voting Power" (VP) in $[0, 1]$, initialized to 1. $a$ and $b$ are system-wide constants that roughly specify how influential a single vote is. 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 $w$ is chosen to be equal to the likability of the post. The functionality increases the score of the post by $\mathrm{SP}\,(a \cdot \mathrm{VP} \cdot w + b)$ and subsequently decreases the player's Voting Power by the same amount (but keeping it within the aforementioned bounds). Voting Power is replenished with time, at a rate defined by the parameter regen. The purpose of Voting Power is to "rate limit" votes.

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

▶ Remark 10. 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 $\left\lceil \frac{a+b}{\mathrm{regen}} \right\rceil > 1$ 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 11. The Steem blockchain protocol defines $a = 0.02, b = 0.0001$ and $\mathrm{regen} = \frac{3}{5 \cdot 24 \cdot 60 \cdot 60} = 0.00000694\bar{4}$, thus $\left\lceil \frac{a+b}{\mathrm{regen}} \right\rceil = 2895$. A post can be voted for 7 days from its creation and at most one vote can be cast every 3 seconds, thus $R = \frac{7 \cdot 24 \cdot 60 \cdot 60}{3} = 201600$. We do not know why these particular parameters were chosen, but we conjecture that $a, b$ and regen ensure users can vote often enough without abusing the system, 7 days is the time needed for the quality of a post to be determined and 3 seconds is the time needed for transactions to settle in the Steem blockchain.

▶ Remark 12. 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 reward [18].

▶ **Theorem 13.**

1. *If* $\exists i \neq j \in [N] : \mathrm{SP}_i \neq \mathrm{SP}_j$ *(i.e. if not all players have the same Steem Power) then Steemit does not* $(N, R, M, 1)$*-converge.*
2. *If* $\forall i \neq j \in [N], \mathrm{SP}_i = \mathrm{SP}_j$ *(i.e. if all players have the same Steem Power) and*
   **a.** $R - 1 \geq (M - 1) \left\lceil \frac{a+b}{\mathrm{regen}} \right\rceil$ *then Steemit* $(N, R, M, M)$*-converges.*

**b.** $R-1 < (M-1)\left\lceil \frac{a+b}{regen} \right\rceil$ *then Steemit does not* $(N, R, M, 1)$*-converge.*

*Proof Sketch.* When **SP** is not constant, we build a post list where the most liked post is not preferred by rich players and thus is not placed at the top. For a constant **SP**, 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. ◀

See Appendix A for proof.

▶ **Corollary 14.** *The Steemit system parametrised according to Remark 11, for any number of players* $N \geq 2$*, constant **SP** and* $M \leq 70$ *posts* $(N, R, M, M)$*-converges. If* $M > 70$ *or **SP** is not constant, then there exists a list of posts such that the system does not* $(N, R, M, 1)$*-converge.*

## 4 Simulation

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 13, which implies that the system converges perfectly when a sufficient number of voting rounds is permitted, but otherwise the resulting list of posts may have a 0-ideal rank, i.e. the top post may not have the best ideal score. Moreover, we compare our $t$-convergence metric with previously used metrics of convergence based on correlation demonstrating that they are very closely aligned.

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 [10] 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 even a single selfish player has a detrimental effect to the $t$-ideal rank 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.

### 4.1 Methodology

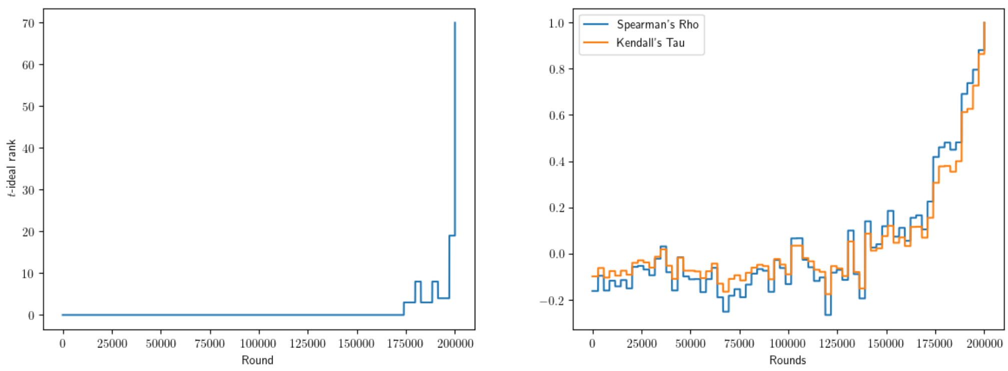
We leverage three metrics to compare the curated list with the ideal list: Kendall's Tau [25], Spearman's Rho [35], and $t$-ideal rank.

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.

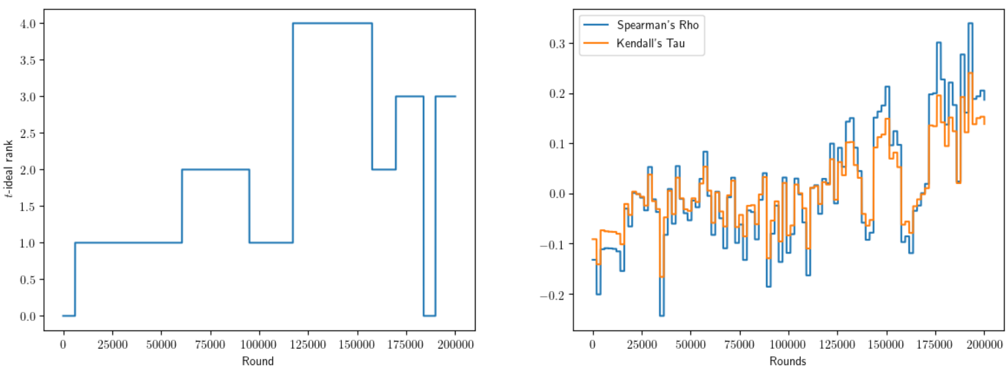
## 4.2 Execution

In all simulations, the likabilities of all "honest" posts have been drawn from the $[0, 1]$-uniform distribution and all players have Steem Power equal to 1; we leave the case of variable Steem Power as future work.




**(a)** $t$-ideal rank evolution **(b)** Kendall's Tau and Spearman's Rho evolution

**Figure 1** 270 honest players, 70 posts and 200.000 rounds

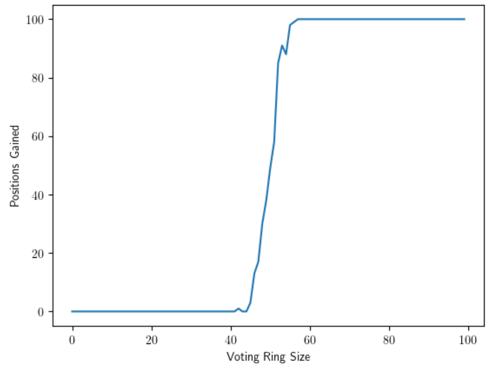



**(a)** $t$-ideal rank evolution **(b)** Kendall's Tau and Spearman's Rho evolution

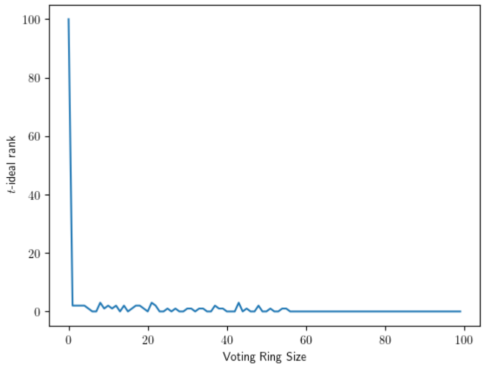
**Figure 2** 300 honest players, 100 posts and 200.000 rounds

### 4.2.1 Scenario A

As already mentioned, the results closely follow Theorem 13. Figures 1a and 1b 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 $R - 1 > (M - 1) \left\lceil \frac{a+b}{\text{regen}} \right\rceil$.)




**(a)** Positions gained by selfish post




**(b)** $t$-ideal rank

**Figure 3** 100 honest players, 100 posts and 0 to 100 selfish players

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

Figures 2a and 2b depict what happens when the rounds are not sufficient for all votes to be cast with full Voting Power. In particular, the corresponding simulation was executed with the same parameters, except for $M = 100$ and $N = 300$. (Observe that $R - 1 < (M - 1) \left\lceil \frac{a+b}{\text{regen}} \right\rceil$.)

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, ending up in a state of bad quality.

### 4.2.2 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 ring sizes, where ring members vote only for a particular "selfish" post. We fix the rest of the system parameters to handicap the selfish post. 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 3a shows the gain of the selfish post for a varying number of selfish players, from 1 to 100. Figure 3b depicts the $t$-ideal rank of the resulting list at the same executions. The system parameters are $N = 101..200, a = \frac{1}{50}, b = 10^{-4}, \text{regen} = \frac{3}{5 \cdot 24 \cdot 60}, \text{attSpan} = 10, R = 5000$.

As we can see in Figure 3a, there is a cutoff point around which the selfish players quickly move from gaining no positions to overtaking all honest posts. The number of selfish players needed for this advantage is approximately half of the amount of honest ones. On the other hand, figure 3b shows that even a single selfish player can almost completely ruin the $t$-ideal rank of the result by only allowing a very small number of the best posts to be placed in the correct order.

## 5 Summary and Future Work

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.

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 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, voting rings can be allowed 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 analysis can be augmented with a review of games with selfish players and voting rings.

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. A dynamic model in which posts can be created at any time and the execution continues indefinitely (as is the case in a real-world UGC system) is also interesting as a future direction.

## A Proof of Theorem 13: Steem Convergence

**Proof.** Statement 1: Reorder the players such that $\mathrm{SP}_1 \geq \mathrm{SP}_2 \geq \cdots \geq \mathrm{SP}_N$. Let $k = \min_{j \in [N-1]} \{\mathrm{SP}_j \neq \mathrm{SP}_{j+1}\}$. We first cover the case when $\mathrm{attSpan} \geq 2$.

Let[8]

$$\mathrm{weakPost} = (\underbrace{0, \dots, 0}_{k-1}, 1, \underbrace{0, \dots, 0}_{N-k})$$

$$\mathrm{strongPost} = (\underbrace{0, \dots, 0}_{k-1}, \frac{\mathrm{SP}_k - \mathrm{SP}_{k+1}}{2\mathrm{SP}_k}, 1, \underbrace{0, \dots, 0}_{N-k-1})$$

$$\mathrm{nullPost} = (\underbrace{0, \dots, 0}_{N})$$

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

We first note that $\mathrm{SP}_k > \mathrm{SP}_{k+1} \geq 0 \Rightarrow 0 \leq \frac{\mathrm{SP}_k - \mathrm{SP}_{k+1}}{2\mathrm{SP}_k} \leq 1$, thus strongPost is a valid

[8] We thank Heng Guo from the University of Edinburgh for this counterexample.

post. We then observe that

$$\forall i \in \{3, \ldots, M\}, \mathrm{idealSc}(\mathcal{P}[i]) = 0 <$$
$$< \mathrm{idealSc}(\mathcal{P}[1]) = 1 < 1 + \frac{\mathrm{SP}_k - \mathrm{SP}_{k+1}}{2\mathrm{SP}_k} = \mathrm{idealSc}(\mathcal{P}[2]) \quad ,$$

thus $\forall \mathcal{P}'$ that contain the same posts as $\mathcal{P}$ and $\textsc{Ideal}^1(\mathcal{P}')$ holds, it is $\mathcal{P}'[1] = \mathcal{P}[2]$. Since attSpan $\geq 2$, all players apart from $u_{k+1}$ vote for $\mathcal{P}[1]$ in the first round and for $\mathcal{P}[2]$ in the second, whereas $u_{k+1}$ votes for $\mathcal{P}[2]$ in the first round and for $\mathcal{P}[1]$ in the second. Thus the two first posts will have been voted by all players by the end of the second round and their score will not change until the execution completes. We have:

$$\mathrm{sc}_2(\mathcal{P}[1]) = \mathrm{sc}_R(\mathcal{P}[1]) =$$
$$\sum_{j=1}^{k-1} \mathrm{SP}_j b + \mathrm{SP}_k(a+b) + \mathrm{SP}_{k+1} \min\left\{b, \mathbf{VP}\mathrm{reg}_{k+1,r_2}\right\} + \sum_{j=k+2}^{M} \mathrm{SP}_j b \text{ and}$$
$$\mathrm{sc}_2(\mathcal{P}[2]) = \mathrm{sc}_R(\mathcal{P}[2]) =$$
$$\sum_{j=1}^{k-1} \mathrm{SP}_j \min\{b, \mathbf{VP}\mathrm{reg}_{j,r_2}\} +$$
$$\mathrm{SP}_k \min\{a\frac{\mathrm{SP}_k - \mathrm{SP}_{k+1}}{2\mathrm{SP}_k}\mathbf{VP}\mathrm{reg}_{k,r_2} + b, \mathbf{VP}\mathrm{reg}_{k,r_2}\} + \mathrm{SP}_{k+1}(a+b) +$$
$$\sum_{j=k+2}^{M} \mathrm{SP}_j \min\{b, \mathbf{VP}\mathrm{reg}_{j,r_2}\} \Rightarrow$$

$$\mathrm{sc}_R(\mathcal{P}[2]) \leq$$
$$\sum_{j=1}^{k-1} \mathrm{SP}_j b + \mathrm{SP}_k(a\frac{\mathrm{SP}_k - \mathrm{SP}_{k+1}}{2\mathrm{SP}_k} + b) + \mathrm{SP}_{k+1}(a+b) + \sum_{j=k+2}^{M} \mathrm{SP}_j b \quad .$$

In the case that $\mathbf{VP}\mathrm{reg}_{k+1,r_2} \geq b$, it is

$$\mathrm{sc}_R(\mathcal{P}[1]) = \sum_{j=1}^{k-1} \mathrm{SP}_j b + \mathrm{SP}_k(a+b) + \mathrm{SP}_{k+1} b + \sum_{j=k+2}^{M} \mathrm{SP}_j b >$$
$$\sum_{j=1}^{k-1} \mathrm{SP}_j b + \mathrm{SP}_k(a\frac{\mathrm{SP}_k - \mathrm{SP}_{k+1}}{2\mathrm{SP}_k} + b) + \mathrm{SP}_{k+1}(a+b) + \sum_{j=k+2}^{M} \mathrm{SP}_j b \geq$$
$$\mathrm{sc}_R(\mathcal{P}[2]) \Rightarrow \mathrm{sc}_R(\mathcal{P}[1]) > \mathrm{sc}_R(\mathcal{P}[2]) \quad ,$$

thus $\textsc{Ideal}^1(\mathcal{P}')$ does not hold.
Since $u_{k+1}$ does not vote in any round between $r_1$ and $r_2$, and $r_2 \geq 2$, it is $\mathbf{VP}\mathrm{reg}_{k+1,r_2} \geq 1 - a - b + \mathrm{regen}$. Thus the case when $\mathbf{VP}\mathrm{reg}_{k+1,r_2} < b$ can happen only when $b > 1 - a - b + \mathrm{regen} \Leftrightarrow b > \frac{1-a+\mathrm{regen}}{2}$. We now provide a counterexample for the case when $b > \frac{1-a+\mathrm{regen}}{2}$.
Once more we order the players in descending Steem Power, like in the previous case. Once again $k = \min_{j \in [N-1]} \{\mathrm{SP}_j \neq \mathrm{SP}_{j+1}\}$ and we only care for the case when attSpan $\geq 2$.

Let $0 < \gamma < 1$ and

$$\text{weakPost} = (\underbrace{0, \ldots, 0}_{k-1}, 1, \frac{\gamma}{2}, \underbrace{0, \ldots, 0}_{N-k-1})$$

$$\text{strongPost} = (\underbrace{0, \ldots, 0}_{k-1}, \gamma, 1, \underbrace{0, \ldots, 0}_{N-k-1})$$

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

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

We observe that $\forall i \in \{3, \ldots, M\}, \text{idealSc}(\mathcal{P}[i]) = 0 < \text{idealSc}(\mathcal{P}[1]) = 1 + \frac{\gamma}{2} < 1 + \gamma = \text{idealSc}(\mathcal{P}[2])$, thus $\forall \mathcal{P}'$ that contain the same posts as $\mathcal{P}$ and $\textsc{Ideal}^1(\mathcal{P}')$ holds, it is $\mathcal{P}'[1] = \mathcal{P}[2]$.
Since $\text{attSpan} \geq 2$, all players apart from $u_{k+1}$ vote for $\mathcal{P}[1]$ in the first round and for $\mathcal{P}[2]$ in the second, whereas $u_{k+1}$ votes for $\mathcal{P}[2]$ in the first round and for $\mathcal{P}[1]$ in the second. Thus the two first posts will have been voted by all players by the end of the second round and their score will not change until the execution completes. We have:

$$\begin{aligned}
&\text{sc}_2(\mathcal{P}[1]) = \text{sc}_R(\mathcal{P}[1]) = \\
&\sum_{j=1}^{k-1} \text{SP}_j b + \text{SP}_k (a+b) + \text{SP}_{k+1} \mathbf{VP}\text{reg}_{k+1,r_2} + \sum_{j=k+2}^{M} \text{SP}_j b \text{ and} \\
&\text{sc}_2(\mathcal{P}[2]) = \text{sc}_R(\mathcal{P}[2]) = \\
&\sum_{j=1}^{k-1} \text{SP}_j \min\{b, \mathbf{VP}\text{reg}_{j,r_2}\} + \text{SP}_k \mathbf{VP}\text{reg}_{k,r_2} + \text{SP}_{k+1}(a+b) + \\
&\sum_{j=k+2}^{M} \text{SP}_j \min\{b, \mathbf{VP}\text{reg}_{j,r_2}\} \leq \\
&\sum_{j=1}^{k-1} \text{SP}_j b + \text{SP}_k \mathbf{VP}\text{reg}_{k,r_2} + \text{SP}_{k+1}(a+b) + \sum_{j=k+2}^{M} \text{SP}_j b \ .
\end{aligned}$$

We note that $\mathbf{VP}\text{reg}_{k,r_2} = \mathbf{VP}\text{reg}_{k+1,r_2}$ because both $u_k$ and $u_{k+1}$ vote with full Voting Power in the first round. Let $\text{VP} = \mathbf{VP}\text{reg}_{k,r_2}$. We have

$$\begin{aligned}
&\text{SP}_k(a+b) + \text{SP}_{k+1}\text{VP} > \text{SP}_k\text{VP} + \text{SP}_{k+1}(a+b) \Leftrightarrow \\
&\text{SP}_k(a+b) + \text{SP}_{k+1}\text{VP} - \text{SP}_k\text{VP} - \text{SP}_{k+1}(a+b) > 0 \Leftrightarrow \\
&(a+b)(\text{SP}_k - \text{SP}_{k+1}) - \text{VP}(\text{SP}_k - \text{SP}_{k+1}) > 0 \Leftrightarrow \\
&(\text{SP}_k - \text{SP}_{k+1})(a+b-\text{VP}) > 0
\end{aligned}$$

The last expression is true because $\text{SP}_k > \text{SP}_{k+1}$ and $\text{VP} < b$, thus the first expression is true as well. We can then deduce that $\text{sc}_R(\mathcal{P}[1]) > \text{sc}_R(\mathcal{P}[2])$, thus $\textsc{Ideal}^1(\mathcal{P}')$ does not hold. Please refer to the full version [26] for the case when $\text{attSpan} = 1$.

- Statement 2a: Suppose that

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

Observe that

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

Let pid $\in [N]$. From (1) we deduce that $R \geq M$ and according to VoteThisRound in Algorithm 6, $u_{\text{pid}}$ votes non-null in rounds $(r_1, \dots, r_M)$ with $r_i = \left\lfloor (i-1) \frac{R-1}{M-1} \right\rfloor + 1$. We define the following:

$$k \in \mathbb{N}, w \in \mathbb{R} \ ,$$
$$n \in \mathbb{Z}, p \in [0,1) : (k-1)\, w = n + p \ ,$$
$$m \in \mathbb{Z}, q \in [0,1) : w = m + q \ .$$

We have

$$\lfloor (k-1)\, w \rfloor = n \ , \tag{3}$$

$$\lfloor kw \rfloor = \begin{cases} n+m, & p+q<1 \\ n+m+1, & p+q \geq 1 \text{ (impossible if } p=0) \end{cases} \tag{4}$$

$$\lfloor w \rfloor = m \tag{5}$$

$$\lceil w \rceil = \begin{cases} m, & p=0 \\ m+1, & p>0 \end{cases} \tag{6}$$

$$\begin{gathered} (3), (4), (5), (6), p+q<2 \Rightarrow \\ \lfloor kw \rfloor \in \{ \lfloor (k-1)\, w \rfloor + \lfloor w \rfloor, \lfloor (k-1)\, w \rfloor + \lceil w \rceil \} \end{gathered} \tag{7}$$

From (7) we deduce that

$$\forall i \in [M] \setminus \{1\}, r_i \in \{ r_{i-1} + \left\lfloor \frac{R-1}{M-1} \right\rfloor, r_{i-1} + \left\lceil \frac{R-1}{M-1} \right\rceil \} \ . \tag{8}$$

From (2) and (8) 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

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

  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

$$\forall i \neq j \in [M], \text{idealSc}(\mathcal{P}[i]) > \text{idealSc}(\mathcal{P}[j]) \Rightarrow$$
$$\sum_{\text{pid}=1}^{N} \mathcal{P}[i]_{\text{pid}} > \sum_{\text{pid}=1}^{N} \mathcal{P}[j]_{\text{pid}} \Rightarrow$$
$$Nb + a \sum_{\text{pid}=1}^{N} \mathcal{P}[i]_{\text{pid}} > Nb + a \sum_{\text{pid}=1}^{N} \mathcal{P}[j]_{\text{pid}} \ .$$

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

Statement 2b: Suppose that

$$R - 1 < (M-1)\left\lceil \frac{a+b}{\text{regen}} \right\rceil \ . \tag{9}$$

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

$$\text{attSpan} + R \leq M \ . \tag{10}$$

In this case, no player can ever vote for the last post, as we will show now. First of all, (10) $\Rightarrow R < M$, thus all players cast $R$ votes in total. Let $\text{pid} \in N, i \in [R]$ 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 $\forall \text{pid} \in N, v_{\text{pid},R} = \text{attSpan} + R - 1 \overset{(10)}{<} M$. 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

$$\text{strongPost} = (\underbrace{1, \dots, 1}_{N}), \text{nullPost} = (\underbrace{0, \dots, 0}_{N})$$
$$\mathcal{P} = [\underbrace{\text{nullPost}, \dots, \text{nullPost}}_{M-1}, \text{strongPost}]$$

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

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

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

i.e. $i$ is the first round in which $u_{\text{pid}}$ 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 $i \geq 2$.

- If $R \geq M$, it is $V = M$.
If $\nexists i \in [M] : \left(\mathbf{VP}\mathrm{reg}_{\mathrm{pid},r_i} < 1 \wedge \nexists i' < i : \mathbf{VP}\mathrm{reg}_{\mathrm{pid},r_{i'}} < 1\right)$, then we have that $\forall i \in [M], \mathbf{VP}\mathrm{reg}_{\mathrm{pid},r_i} = 1 \Rightarrow \forall i \in [M] \setminus \{1\}, r_i \geq r_{i-1} + \left\lceil \frac{a+b}{\mathrm{regen}} \right\rceil$ 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

$$\left\lceil \frac{a+b}{\mathrm{regen}} \right\rceil \geq 2 \Rightarrow \frac{a+b}{\mathrm{regen}} > 1 \Rightarrow a+b > \mathrm{regen} \ . \tag{11}$$

Thus $\forall \mathrm{pid} \in [N], \mathbf{VP}\mathrm{reg}_{\mathrm{pid},r_2} = 1 - a - b + \mathrm{regen} \overset{(11)}{<} 1$, 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 $\mathrm{rVP} = \mathbf{VP}\mathrm{reg}_{1,r_i}$. Assume that $\mathrm{attSpan} < i \vee i > 2$. Please refer to the full version [26] for the case when $\mathrm{attSpan} \geq i \wedge i = 2$. In case $N$ is even, let $0 < \gamma < 0, 0 < \epsilon < \gamma(1 - \mathrm{rVP})$,

$$\mathrm{weakPost} = (\underbrace{1, \dots, 1}_{N/2}, \underbrace{\gamma - \epsilon, \dots, \gamma - \epsilon}_{N/2}) \ ,$$
$$\mathrm{strongPost} = (\underbrace{\gamma, \dots, \gamma}_{N/2}, \underbrace{1, \dots, 1}_{N/2}), \mathrm{nullPost} = (\underbrace{0, \dots, 0}_{N}) \ ,$$
$$\mathcal{P} = [\underbrace{\mathrm{weakPost}, \dots, \mathrm{weakPost}}_{i-1}, \mathrm{strongPost}, \underbrace{\mathrm{nullPost}, \dots, \mathrm{nullPost}}_{M-i}] \ .$$

First of all, it is

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

and $\forall j \in \{i+1, \dots, M\}, \mathrm{idealSc}(\mathcal{P}[j]) = 0 < \mathrm{idealSc}(\mathcal{P}[i])$, 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 $\textsc{Ideal}^1(\mathcal{P}')$ holds, it is $\mathcal{P}'[1] = \mathcal{P}[i]$.
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 $\mathrm{attSpan} > 1$ 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 $\text{attSpan} = 1$ 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 likability 1, thus in round $r_i$ their Voting Power after regeneration is exactly the same as in the case of $\mathcal{P}^1 : \forall \text{pid} \in \left[\frac{N}{2}\right], \mathbf{VP}\text{reg}_{\text{pid},r_i} = \text{rVP}$.
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 $\text{attSpan} < i$, they all vote for the first weak post of the list in the first round, and thus with full Voting Power. If $\text{attSpan} \geq i$ 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 $\text{sc}_R(\mathcal{P}[1]) = \frac{N}{2}(a+b) + \frac{N}{2}((\gamma - \epsilon)a + b)$.
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 $\text{sc}_R(\mathcal{P}[i]) \leq \frac{N}{2}(\text{rVP} \cdot \gamma a + b) + \frac{N}{2}(a+b)$. It is

$$\begin{aligned}
&\epsilon < \gamma(1 - \text{rVP}) \Rightarrow \\
&\frac{N}{2}(\text{rVP} \cdot \gamma a + b) + \frac{N}{2}(a+b) < \frac{N}{2}(a+b) + \frac{N}{2}((\gamma - \epsilon)a + b) \Rightarrow \\
&\text{sc}_R(\mathcal{P}[i]) < \text{sc}_R(\mathcal{P}[1]) \quad .
\end{aligned}$$

Thus $\mathcal{P}_R[1] \neq \mathcal{P}[i]$ and $\text{Ideal}^1(\mathcal{P}_R)$ does not hold.
As for the case when $N$ is odd, let $0 < \epsilon < \gamma\frac{N-3}{N-1}(1 - \text{rVP})$. In this case, we assume that the likability of the first $i$ posts (weak and strong) for the additional player is $\gamma$, whereas the likability of the last $M - i$ posts (the null posts) is 0. This means that the additional 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 additional 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:

$$\begin{aligned}
&\text{sc}_R(\mathcal{P}[i]) \leq \frac{N-1}{2}(\text{rVP} \cdot \gamma a + b) + \frac{N-1}{2}(a+b) + \gamma a + b \ , \\
&\text{sc}_R(\mathcal{P}[1]) \geq \frac{N-1}{2}(a+b) + \frac{N-1}{2}((\gamma - \epsilon)a + b) + \text{rVP} \cdot \gamma a + b \ .
\end{aligned}$$

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

◀

# B Steem Post Voting System Procedures

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

Store input parameters as constants
$r \leftarrow 1$
$\text{lastVoted} \leftarrow (0, \ldots, 0) \in (\mathbb{N}^*)^N$
$\mathbf{VP} \leftarrow (1, \ldots, 1) \in [0,1]^N$
$\text{scores} \leftarrow (0, \ldots, 0) \in (\mathbb{R}^+)^M$

**Algorithm 4** AUX

**return** $(\text{attSpan}, a, b, r, \text{regen}, R, \mathbf{SP})$

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

**if** $\text{lastVoted}_{\text{pid}} \neq r$ **then** ▷ One vote per player per round
    $\mathbf{VP}_{\text{pid},r} \leftarrow \mathbf{VP}_{\text{pid}}$ ▷ For proofs
    $\mathbf{VP}_{\text{pid}} \leftarrow \max\{\mathbf{VP}_{\text{pid}} + \text{regen}, 1\}$
    $\mathbf{VP}\text{reg}_{\text{pid},r} \leftarrow \mathbf{VP}_{\text{pid}}$ ▷ For proofs
    **if** ballot $\neq$ **null then**
        Parse ballot as $(P, \text{weight})$
        $\text{cost} \leftarrow a \cdot \mathbf{VP}_{\text{pid}} \cdot \text{weight} + b$
        **if** $\mathbf{VP}_{\text{pid}} - \text{cost} \geq 0$ **then**
            $\text{score} \leftarrow \text{cost} \cdot \mathbf{SP}_{\text{pid}}$
            $\mathbf{VP}_{\text{pid}} \leftarrow \mathbf{VP}_{\text{pid}} - \text{cost}$
        **else**
            $\text{score} \leftarrow \mathbf{VP}_{\text{pid}} \cdot \mathbf{SP}_{\text{pid}}$
            $\mathbf{VP}_{\text{pid}} \leftarrow 0$
        **end if**
        $\text{scores}_P \leftarrow \text{scores}_P + \text{score}$
    **end if**
    $\text{lastVoted}_{\text{pid}} \leftarrow r$
**end if**
**if** $\forall i \in [N], \text{lastVoted}_i = r$ **then** ▷ round over
    $\mathcal{P} \leftarrow \text{ORDER}(\mathcal{P}, \text{scores})$ ▷ order posts by votes
    $\mathcal{P}_r \leftarrow \mathcal{P}$ ▷ For proofs
    $r \leftarrow r + 1$
**end if**


OASICS OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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

```
Store aux contents as constants
voteRounds ← VoteRounds (R, |𝒫|)
if VoteThisRound (r, |𝒫|) = yes then
    top ← ChooseTopPosts (attSpan, 𝒫, votedPosts)
    (i, l) ← argmax_{(i,l)∈top} {l_pid}[1]
    votedPosts ← votedPosts ∪ (i, l)
    return ((i, l), l_pid)
else
    return null
end if

function ChooseTopPosts(attSpan, 𝒫, votedPosts)
    res ← ∅
    idx ← 1
    while |res| < attSpan & idx ≤ |𝒫| do
        if 𝒫[idx] ∉ votedPosts then                ▷ One vote per post per player
            res ← res ∪ {𝒫[idx]}
        end if
        idx ← idx + 1
    end while
    return res
end function

function VoteThisRound(r, M)
    if R < M then
        return yes
    else if r ∈ voteRounds then
        return yes
    else
        return no
    end if
end function

function VoteRounds(R, M)
    voteRounds ← ∅
    for i = 1 to M do
        voteRounds ← voteRounds ∪ {1 + ⌊(i − 1) (R−1)/(M−1)⌋}
    end for
    return voteRounds
end function
```

## References


1 Zeinab Abbassi, Nidhi Hegde, and Laurent Massoulié. Distributed content curation on the web. *ACM Transactions on Internet Technology (TOIT)*, 14(2-3):9, 2014.

2 Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Steering user behavior with badges. In *Proceedings of the 22nd international conference on World Wide*





OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

*Web*, pages 95–106. ACM, 2013.

**3** Georgios Askalidis and Greg Stoddard. A theoretical analysis of crowdsourced content curation. In *The 3rd Workshop on Social Computing and User Generated Content*, volume 16, 2013.

**4** Kelly Bergstrom. "don't feed the troll": Shutting down debate about community expectations on reddit. com. *First Monday*, 16(8), 2011.

**5** Vitalik Buterin. Notes on blockchain governance. 2017. `https://vitalik.ca/general/2017/12/17/voting.html` Accessed: 2019-01-02.

**6** Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. In *Proceedings of the 6th ACM conference on Electronic commerce*, pages 78–87. ACM, 2005.

**7** Philip Daian, Tyler Kell, Ian Miers, and Ari Juels. On-chain vote buying and the rise of dark daos. 2018. `http://hackingdistributed.com/2018/07/02/on-chain-vote-buying/` Accessed: 2019-01-02.

**8** dantheman. Dpos consensus algorithm - the missing white paper. 2017. `https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper` Accessed: 2019-04-02.

**9** Anish Das Sarma, Atish Das Sarma, Sreenivas Gollapudi, and Rina Panigrahy. Ranking mechanisms in twitter-like forums. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 21–30. ACM, 2010.

**10** J. R. Douceur. The sybil attack. *International workshop on Peer-To-Peer Systems*, 2002.

**11** Evan Duffield and Daniel Diaz. Dash: A privacycentric cryptocurrency. *Self-published*, 2015.

**12** Fred Ehrsam. Blockchain governance: Programming our future. *https://www.medium.com*, 2017. `https://www.medium.com/@FEhrsam/blockchain-governance-programming-our-future-c3bfe30f2d74` Accessed: 2019-01-02.

**13** Andrea Forte and Amy Bruckman. Scaling consensus: Increasing decentralization in wikipedia governance. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pages 157–157. IEEE, 2008.

**14** Arpita Ghosh and Preston McAfee. Incentivizing high-quality user-generated content. In *Proceedings of the 20th international conference on World wide web*, pages 137–146. ACM, 2011.

**15** Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica: journal of the Econometric Society*, pages 587–601, 1973.

**16** Mike Goldin. Token-curated registries 1.0. 2017. `https://medium.com/@ilovebagels/token-curated-registries-1-0-61a232f8dac7` Accessed: 2019-01-02.

**17** Oded Goldreich. The foundations of modern cryptography. In *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*, pages 1–37. Springer, 1999.

**18** Julián González. Author and curator rewards in hf19. 2018. `https://steemit.com/steemit/@jga/author-and-curator-rewards-in-hf19`. Accessed: 2019-01-02.

**19** Julián González. Self-voters can achieve an interest of 248% apr!! 2018. `https://steemit.com/utopian-io/@jga/self-voters-can-achieve-an-interest-of-248-apr`. Accessed: 2019-01-02.

**20** LM Goodman. Tezos–a self-amending crypto-ledger white paper. 2014. `https://www.tezos.com/static/papers/white_paper.pdf`.

**21** Mangesh Gupte, MohammadTaghi Hajiaghayi, Lu Han, Liviu Iftode, Pravin Shankar, and Raluca M Ursu. News posting by strategic users in a social network. In *International Workshop on Internet and Network Economics*, pages 632–639. Springer, 2009.

**22** Claude Hillinger. The case for utilitarian voting. *Homo Oeconomicus*, 22(3), 2005. `https://ssrn.com/abstract=878008`. Accessed: 2019-04-01.

**23** Meir Kalech, Sarit Kraus, Gal A Kaminka, and Claudia V Goldman. Practical voting rules with partial information. *Autonomous Agents and Multi-Agent Systems*, 22(1):151–182, 2011.

**24** Andreas M Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010.

**25** Maurice G Kendall. *Rank Correlation Methods*, volume 9, page 68. Hafner Publishing Co., 2 edition, 1955. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2044-8317.1956.tb00172.x. Accessed: 2019-04-01. doi:10.1111/j.2044-8317.1956.tb00172.x.

**26** Aggelos Kiayias, Benjamin Livshits, Andrés Monteoliva Mosteiro, and Orfeas Stefanos Thyfronitis Litos. A puff of steem: Security analysis of decentralized content curation. *arxiv.org*, 2018.

**27** Dor Konforty, Yuval Adam, Daniel Estrada, and Lucius Gregory Meredith. Synereo: The decentralized and distributed social network. *Self-published*, 2015. https://pdfs.semanticscholar.org/253c/c4744e6b2b87f88e46188fe527982b19542e.pdf Accessed: 2019-01-02.

**28** Jure Leskovec, Daniel P Huttenlocher, and Jon M Kleinberg. Governance in social media: A case study of the wikipedia promotion process. In *ICWSM*, pages 98–105, 2010.

**29** Brian Neil Levine, Clay Shields, and N Boris Margolin. A survey of solutions to the sybil attack. *University of Massachusetts Amherst, Amherst, MA*, 7:224, 2006.

**30** Yehuda Lindell and Jonathan Katz. *Introduction to modern cryptography*. Chapman and Hall/CRC, 2014.

**31** Tyler Lu and Craig Boutilier. Robust approximation and incremental elicitation in voting protocols. In *IJCAI*, volume 1, pages 287–293, 2011.

**32** Avner May, Augustin Chaintreau, Nitish Korula, and Silvio Lattanzi. Filter & follow: How social media foster content curation. In *ACM SIGMETRICS Performance Evaluation Review*, volume 42, pages 42–55. ACM, 2014.

**33** Emilee Rader and Rebecca Gray. 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*, pages 173–182. ACM, 2015.

**34** Fabian Schuh. Graphene documentation. 2018. https://media.readthedocs.org/pdf/docsbitsharesorg/master/docsbitsharesorg.pdf Accessed: 2019-04-02.

**35** Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904.

**36** Katarina Stanoevska-Slabeva, Vittoria Sacco, and Marco Giardina. Content curation: a new form of gatewatching for social media? *Documento electrónico. Recuperado el*, 16, 2012.

**37** Mike Thelwall. Can social news websites pay for content and curation? the steemit cryptocurrency model. *Journal of Information Science*, page 0165551517748290, 2017.

**38** Unknown. Steem: An incentivized, blockchain-based, public content platform. 2017. https://steem.com/SteemWhitePaper.pdf Accessed: 2019-04-02.

**39** Unknown. Steem whitepaper. 2018. https://steem.io/steem-whitepaper.pdf Accessed: 2019-04-02.

**40** Lirong Xia and Vincent Conitzer. Compilation complexity of common voting rules. In *AAAI*, 2010.