Analysis and Attacks of decentralized content curation platforms

Andrés Monteoliva Mosteiro and Orfeas Stefanos Thyfronitis Litos

University of Edinburgh
o.thyfronitis@ed.ac.uk, a.monteoliva@serious.server

Abstract. We will attack Steem.

1 Introduction

Steem is not incentive-compatible.

2 Related Work

Many people have done many similar things.

3 Model

1 Properties

Players The are $N \in \mathbb{N}^*$ players in the model. A player u is defined by her Steem Power $SP \in \mathbb{N}$, her Voting Power $VP \in [0,1]$, her Likeability Distribution $L \in \mathcal{D}\left([0,1]^N\right)$ and her Strategy $S \in \{H, G\} \times \times 2^{[N]}$.[1] Let the ith player be represented by the tuple $u_i = (SP_i, VP_i, L_i, S_i)$. The tuple of players is defined as $\mathcal{U} = (u_1, \dots, u_i, \dots, u_N)$ and each player is identified by her index in $\mathcal{U}$.

We will now explain each field in u_i in detail:

- **Steem Power.** The Steem Power of u_i is defined as $SP_i \in \mathbb{N}$ and represent the influence of the player in the platform. The vector of Steem Power funds for the N players is defined as $\mathcal{SP} = (SP_1, \dots, SP_i, \dots, SP_n)$.
- **Voting Power.** The Voting Power of u_i is defined as $VP_i \in [0,1]$ and can be understood as voting influence that is used up when voting and regenerates with time. The vector of Voting Power for the N players is defined as $\mathcal{VP} = (VP_1, \dots, VP_i, \dots, VP_N)$.

[1] We denote the powerset of a set A with 2^A.

- **Likeability Distribution.** The Likeability Distribution $L_i \in \mathcal{D}\left([0,1]^N\right)$[2] of u_i is a distribution on how likeable is the content produced by u_i to the rest of the players. The Likeability Distribution for the whole system is $\mathcal{L} = (L_1, \dots, L_i, \dots, L_n)$.
- **Strategy.** The strategy of u_i is defined as $S_i \in \{H, G\} \times \mathbb{N}^* \times 2^{[N]}$, where the first element is the player's core strategy, the second is her attention span and the third is her voting ring.
  - **Honest/Greedy.** H corresponds to the *honest* and G to the *greedy* strategy. An *honest* player votes according to the likeability of a post l_i (defined later)[3], that is to say she votes the posts she likes. For *honest* players, the value of the vote is computed as $v_{H,i} = VP \cdot l \cdot SP$, where l is drawn from the Likeability distribution[4]. In Steem terms, l can be understood as the weight of a vote.

    A *greedy* player only votes for posts produced by users of its Voting Ring. The value of vote for a player if u_i is *greedy* is defined as $v_{G,i} = VP_i \cdot SP_i$,[5] as in our model all *greedy* votes are executed with full weight.
  - **Attention Span.** This is a positive integer that represents the number of posts a player can consider voting simultaneously. For the benefit of simplicity, we will assume that this number is constant throughought all players.[6]
  - **Voting Ring.** If player u_i is *honest*, her Voting Ring is $R_i = \emptyset$. If u_i is *greedy*, her Voting Ring is $R_i \in 2^{[N]}$. A voting ring is defined as $R_i = \{g_1, \dots, g_j, \dots, g_n\}$ where $g_j \in \mathcal{U}$ is the jth member of the voting ring and n is the size of the ring. Two *greedy* players will either have the same or disjoint voting rings $(\forall i \neq j \in [N], R_i = R_j \vee (R_i \cap R_j = \emptyset))$.

  The tuple of the strategies for the N players is defined as $\mathcal{S} = (S_1, \dots, S_i, \dots, s_N)$.

The set of players is defined as $\mathcal{U} = (u_1, ..., u_i, ..., u_n)$ $\forall i \in [N]$.[7]

Posts A post is defined as $p = (i, l, v)$, with $i \in [N], l \sim L_i$ and $v \in \mathbb{R}_+$.

- **Author.** Each player u_i creates one post. The first element of that post is the index of its creator, i.

[2] We denote the set of all probability distributions on set A as $\mathcal{D}(\mathcal{A})$.

[3] TODO: untangle

[4] TODO: move to posts section

[5] TODO: same

[6] TODO: discuss

[7] TODO: fix appearance

- **Likeability.** The likeability of a post is defined as $l \in [0,1]^N$, where l is drawn from L_i (the Likeability Distribution of its creator u_i).
- **Votes.** A post has an associated "vote" value, which is a real non-negative number. It is initialized at 0 and increases whenever a player votes for the post, as explained later in detail.

Let $l_i \sim L_i$ and $p_i = (i, l_i, 0)$. The set of all posts is $\mathcal{P} = \|_{i=1}^{N} p_i$.[8]

2 Game Execution

Algorithm 1 Each player creates one post

```
function GenerratePosts($\mathcal{U}$)
    $\mathcal{P} = \emptyset$                                         ▷ List of posts
    for $u_i \in \mathcal{U}$ do
        $l \xleftarrow{r} L_i$                                    ▷ Get likeability of posts
        $\mathcal{P} \leftarrow \mathcal{P} \| (i, l, 0)$                         ▷ Add post to list of Posts
    end for
    $\mathcal{P} \leftarrow$ Shuffle($\mathcal{P}$)                   ▷ Shuffle posts to a random order
    return $\mathcal{P}$
end function
```

Algorithm 3 Players cast votes for r rounds

```
function Curate($\mathcal{U}, \mathcal{P}, r$)
    for $j = 1$ to $r$ do                                       ▷ $r$ voting rounds
        for $u_i \in \mathcal{U}$ do
            if IsVoteRound($r, j, S_i$) then
                $\mathcal{P} \leftarrow$ Vote ($u_i, \mathcal{P}$)            ▷ Player $i$ votes zero or one posts
            end if
        end for
        $\mathcal{P} \leftarrow$ Order ($\mathcal{P}$)      ▷ Order posts by vote count after each round of votes
    end for
    return $\mathcal{P}$
end function
```

Algorithm 4 Calculates whether voting in this round is optimal

```
function IsVoteRound($r, j, S$)

end function
```

[8] $a \| b$ denotes the concatenation of a and b.

Algorithm 2 Player votes for best of k posts

```
function VOTE($u_i, \mathcal{P}$)
    switch $S_i$ do
        case $honest$
            $p_j \leftarrow \underset{p \in \mathcal{P}_{1..k}}{\text{argmax}} \{l_{i,p}\}$[9]
            Parse $p_j$ as $(m, l_p, v)$
            $v' \leftarrow v + VP_i \cdot l_{i,p} \cdot sp_i$
            $\mathcal{P} \leftarrow p_1 \| p_2 \| \ldots \| p_{j-1} \| (m, l_p, v') \| p_{j+1} \| \ldots \| p_N$
        end case
        case $Greedy$
                                ▷ If post belongs to voting ring and not reached min VPower
            if $p \in s.R \wedge p.VPower > s.Min$ then
                $voteValue \leftarrow p.VPower \cdot weight \cdot sp$
                $p \leftarrow p.votes + voteValue$
            end if
        end case
    end switch
    return $\mathcal{P}$
end function
```

Algorithm 5 Posts curation procedure

```
    Input: $\mathcal{U}, r$
    Output: $\mathcal{P}$
$\mathcal{P} \leftarrow$ GENERATEPOSTS $(\mathcal{U})$
$\mathcal{P} \leftarrow$ CURATE $(\mathcal{U}, \mathcal{P}, r)$
return $\mathcal{P}$
```

4 Results

Steem won't achieve high quality posts.

5 Further Work

Posts at any time

6 Conclusion

Keep inventing new decentralized content curation platforms.

7 Acknowledgements

We thank Prof. Aggelos Kiayias for constructive conversations, @seriousposter for their invaluable posts analyzing Steem and our mums for the cookies.

References