The main problem with majority voting, as summarized by@vbuterin here, is that voters are not individually accountable: If a vote leads to a bad outcome, there is no way in which someone who voted for that outcome suffers more than someone who voted against.

Futarchy attempts to solve this problem, but most seem to consider that “pure” futarchy is impractical. We disagree with that assessment, and would like to share with the community a setup that we believe solves some known problems. We plan on implementing it as the governance proceduce for our future Tick Exchange DAO.

The DAO issues TICK tokens, while ETH is the base currency. We can only evaluate a single proposal at a time, in what is called a decision round

. Each round can have several decision turns

, which for instance may last for a day. For the duration of a full decision round one can:

- split ETH into ACCEPTED_ETH and DENIED_ETH
- split TICK into ACCEPTED_TICK and DENIED_TICK

At the end of the round a proposal will be either accepted or denied. If a proposal is accepted, ACCEPTED_ETH is converted into ETH and ACCEPTED_TICK being converted TICK. Otherwise, DENIED_ETH is converted into ETH and DENIED_TICK is converted into TICK.

We begin each round by auctioning off some number X

of ACCEPTED_TICK for TICK, and the winner gets to decide what the proposal is for that round. Some proportion u

of the TICK proceeds are then used alongside uX

ACCEPTED_TICK to create a prediction liquidity pool.

This means that the proposal most likely to pass is expected to be decided upon first, and that if the proposal is approved, a total of $(1+p)X$

TICK tokens will be minted.

Once a proposal has been selected for evaluation, the DAO removes some of the liquidity that it holds in a TICK/ETH pool, splits TICK and ETH as described above and adds liquidity to both ACCEPTED_TICK/ACCEPTED_ETH \text{(accepted price)}

and DENIED_TICK/DENIED_ETH \text{(denied price)}

pools.

At the end of each turn, if \text{(accepted price)} < (1 + \alpha)\text{(denied price)}

on average over the turn, then the proposal has a chance p

of being rejected. Otherwise a \text{counter}

is increased, and if the \text{counter} >= N

already, it has a chance q

of being accepted. Once a turn ends with a proposal being either denied or accepted, the round ends.

This proposal is similar to one previously proposed here:Possible Futarchy Setups. However, it solves an important problem out by @vbuterin:

Votes as buy orders: a new type of hybrid coin voting / futarchy

if some vote has a too-close-to-zero chance of passing, there’s no incentive to commit capital to voting against it, and so in equilibrium it has to succeed at least some of the time.

By having probabilistic outcomes for each round, it is possible for an “underdog” proposal to not be denied immediately, giving time for its supporters to sway market opinion. That should prevent prediction markets from estimating approval probabilities that too extreme (very close to either 0 or 100%).

In particular, a proposal has to “pass” several turns to be approved. So when a proposal that was previously considered to be unlikely to pass manages to “pass” one or more rounds, the prediction markets will have time to reevaluate, and token holders that may not be paying attention to it will have time to do so.

Problem #1

- Hostile Takeovers

Someone can make a proposal that effecively steals all the assets from the DAO. If the previous market price of the TICK token is P

, the attacker can bid $P' > P$

in the accepted price

market. Unless the market price of TICK rises after this attack, everyone has to sell their APPROVED_TICK tokens for P'

, in what is effectively a hostile takeover, otherwise their tokens will be worthless. If some significant fraction of tokenholders is expected not to sell, the attack can be very profitable even by paying a high price.

But even less ouright malicious proposals are still bad, for instance a hostile merge/acquisition by a larger competitor project that will stiffle competition.

One potential solution to that is making the α

in the $\text{(accepted price)} < (1 + \alpha)\text{(denied price)}$

equation be variable and depend on the “open interest” of the contract. If there are simultaneously many people bidding APPROVED_ETH for APPROVED_TICK and TICK holders willing to sell conditional on approval, then the α

may increase significantly. This protects inactive tokenholders as controversial proposals become less likely to be approved.

One problem with that is that large TICK holders may effectively simulate a large open interest to bring down proposals, effectively giving them veto power. This could be curbed by adding a “settlement fee” that grows alongside α

, making such vetos costly.

Problem #2

- Denial of service

Someone can bid highly for a nonsense proposal as a way to prevent other proposals from being considered. To do so they will have to bid some number of TICK tokens every round.

Because a proportion $(1-u)$

of the TICK proceeds are effectively profits for the DAO in case the proposal is denied, we can use some of these tokens to increase the value of X

for the next round. As a result, we can make X

increase exponentially if all proposals receive high bids but end up being rejected.

@mkoeppelmann, @danrobinson, do you see any other problem with this idea? Would love to receive feedback from the community.