klerosliearoracledraft7mai2018.pdf (341.5 KB)

I've been working on a decentralized oracle to estimate real-number values, to be built on the (decentralized) dispute resolution platform Kleros (disclaimer: I work for Kleros). In Kleros, crowdsourced, randomly-selected jurors are chosen to rule on disputes, where they are incentivized to be coherent which we use as a proxy for honesty in the style of Schellingcoin and Truthcoin. Then this is already a sort of oracle, where jurors are bringing some real-world knowledge (the correct ruling of a dispute) onto the blockchain. We leverage this into an oracle on real-values (so particularly useful for prices) by allowing anyone to submit an interval they think the value is in (we then call this person a respondent), and to the degree that these intervals are incoherent, that is converted into a (series of) disputes. Specifically, you find points where someone's upper bound is lower than someone else's lower bound and you ask the jurors whether the true value is higher or lower than the median of the set of these points of incoherence.

This proposal has the advantage that how precise the ultimate answer is is tuned to how large the intervals of the respondents are. Particularly, if someone has an interest in the oracle outputting a very precise result, they can submit a very small interval, and the output will be in that interval as long as the jurors don't rule that they were wrong. Respondents place a deposit, so they have to calculate to themselves how large an interval they want to submit in terms of whether they have a financial interest in the oracle outputting a precise value, the higher risk of losing a deposit if their interval is very small, and the fact that if they submit a small interval and are ruled right, they get a higher reward drawn from lost deposits from respondents who were ruled wrong.

Also, the number of times Kleros must be called scales with the \log(\text{Resources of wrong respondents})^2

. So even if there are many responses, that only meaningfully delays the result to the degree that they disagree with each other, and we could reasonably expect the execution time to be good enough to be useful for many applications.

If anyone wants to check out the attached draft paper and/or if you have any thoughts or comments, that would be really appreciated.