Motivation

There have been some interesting discussions in the last few weeks about overbidding in the solver competition and its impact on solvers and users. In this post, we propose a criterion to measure overbidding. The goal is to find a criterion we can turn into a social consensus rule and a CIP.

What is overbidding

According to CIP-20, solvers report a number, denominated in ETH, that is called score, and they are ranked according to their scores. The solver with the highest score wins the auction and, in the case of a successful settlement, receives surplus + fees - second_best_score as a payment, capped by the quantity gas_in_eth + 0.01, where gas_in_eth denotes the ETH spent in gas for the settlement. Since in order to be a winner you need to report the highest score, it is easy to see that the winning solver in case of success receives at least

minimum_payment = min{surplus + fees - winning_score, gas_in_eth + 0.01}

as a payment. Note that the actual payment can indeed be equal to the above quantity.

We now define a score to be reasonable

if it is such that it guarantees that the solver will cover the ETH spent in gas for the settlement (denoted as gas_in_eth), in case of a successful settlement onchain.

Definition [Reasonable score]

. A score is reasonable if the corresponding minimum_payment, in case of a successful settlement, covers the ETH spent in gas for the settlement, which means that a score is reasonable if and only if it satisfies

surplus + fees - score >= gas_in_eth

,

or equivalently,

score <= surplus + fees - gas_in_eth

.

We say that a solver overbids

in an auction if its score, in case it wins and executes its solution onchain, is not reasonable. Note that since gas prices fluctuate, accidental small-scale overbidding can happen. For this reason, we relax the notion and define an average case form of it.

Definition [Reasonable bidding]

. Let {A(1), …, A(n)} be a set of n auctions that a solver wins and successfully executes onchain. We say that this solver bids reasonably if

Σ score(i) <= Σ (surplus(i) + fees(i) - gas_in_eth(i))

,

where score(i), surplus(i), fees(i) and gas_in_eth(i) refer to the corresponding quantities in auction A(i).

Social consensus rule on overbidding

We propose the following rule and test: All solvers have to follow a reasonable bidding strategy. If there is a statistically significant deviation from reasonable bidding, i.e.,

1/nΣ *score(i) > 1/n*Σ (surplus(i) + fees(i) - gas_in_eth(i)) + epsilon,

for some tolerance epsilon, a solver breaks the reasonable bidding rule. Solvers who break this rule can be penalized or slashed.

The differences

1/n*(Σ (surplus(i) + fees(i) - gas_in_eth(i)) - Σ score(i))

for the different solvers are shown in gray (for two weeks, 1.8. - 15.8., fromthis dashboard):

[

image

1850×586 43.5 KB

](https://europe1.discourse-cdn.com/business20/uploads/cow/original/1X/e245d937b0a994a963deac75cd74792ee72833cd.png)

Negative values mean overbidding. This means that the solvers Barter, Baseline, Otex, PropellerSwap, Raven, and SeaSolver have been overbidding according to the test defined above, either intentionally or accidentally (the latter can happen due to miscalculations of gas usage and/or gas prices), and would need to change their score submission strategies.

Discussion

- Why create a social consensus rule instead of 'fixing' the mechanism?

Let's try both. Since designing a mechanism with perfect alignment of incentives for users, solvers, and the protocol has proven difficult in the past, using social consensus rules for some part of the mechanism seems a good intermediary step. The proposed rule is compatible with other proposals to improve the mechanism, e.g., this proposal.

Also, solvers have asked in the past for guidelines for score submission. The proposed rule can be used as such a guideline.

- Why not just use the old objective for ranking?

Using the old objective as score is generally not a reward maximizing strategy for solvers. This is because revert risk is not taken into account and the cost term in the old objective is not an accurate estimate for actual execution costs on chain.

- Why not just define a new and better objective?

We might add the option to submit revert risk and execution costs and have the protocol compute a score from that. This will make it easier to participate in the auction for solvers. It does not, however, fix all problems with overbidding since solvers can still under-report revert risk and costs.

Also, since the general direction for the protocol is to simplify maintaining the protocol, moving more responsibilities to solvers is unavoidable. This includes estimating costs.

Open questions

- Is it always possible to follow a reasonable bidding strategy? (We think it is, but feel free to object.)
- The outlined rule ignores reverts. Should we rather use a criterion which can take reverts into account?
- How should the tolerance epsilon for the test be chosen? Should we use a fixed tolerance or a tolerance derived from a statistical test (e.g. a Student's t-test)?
- Are there better approaches to stop overbidding?