Forum Post and Paper by Max Holloway (@max_holloway)

of Xenophon Labs (@XenophonLabs).

tl;dr

- We (Xenophon Labs) conducted research on the solver competition, and this research culminated in a paper (link).
- As has been suggested before by @tbosman, we recommend what we call a quality-adjusted second-price auction (QASPA). QASPA has a number of nice properties, such as truthful solver reporting, pennying resistance, simplification of the solvers' reporting strategy, moderate collusion resistance, and the incentivization of optimal user outcomes.
- QASPA on its own would produce a solver payment schedule with a fat tail, where some batches have immense payments. We prevent this by enforcing a cap (technically, a cap on the payment coming from the utility difference between the top two solutions). We use historical data to estimate that this cap would be met < 1% of the time if we decide to make mean solver payment equal to the mean of today's status quo solver payment.
- The distribution of second-price solver payments suggests that the protocol is likely overpaying solvers today, and the cap parameter can be used as a way for the protocol to lower its payments. Our estimates show that the protocol could decrease its payments by over 25% while keeping the number of batches affected by the cap below 3%.
- We recommend that the protocol use capped QASPA for its solver competition, but we encourage community members to come forward with any questions/concerns regarding this research and its recommendations.

Introduction

The CoW protocol aims to give users the value of their orderflow by conducting a competition for matching and settling the users' order. Multiple orders are grouped together at discrete intervals, then those orders are sent to specialized agents called solvers

. Solvers compete with one another to provide users with the best matching — as measured by a trader surplus

quantity that serves as a proxy for traders' utility. The matching solution that has the highest trader surplus - execution cost

wins and receives a fixed quantity payment.

Although the current mechanism has performed mostly well historically, it has some issues. One of these issues is pennying, whereby solvers can artificially inflate the trader surplus in order to win the batch, thus beating "honest" solvers who would otherwise win the competition, and driving solver profits to zero. This mechanism also requires the protocol to estimate gas costs, since determining the winner of the auction depends both on gas costs and trader surplus. Furthermore, there is no incentive for solvers to do more than marginally better than the next best solver, since any improvements above that of the next best solver are not rewarded. In our research, we aimed to solve these problems by designing a solver competition that incentivizes solvers to produce excellent solutions and aligns them with the best interest of the protocol and its users. To achieve this, we turn to auctions.

Why auctions?

Well-designed auctions enable an unsophisticated party that is endowed with resources to solicit information and value from a host of more sophisticated parties. In the case of the CoW protocol, the solver competition is run by a protocol that itself is ignorant of optimal order routing, and it solicits optimal routing information from solvers in order to drive value to itself, in the form of increased trader surplus. In return for this information, solvers are rewarded with payment. If we assume that these solvers aim to maximize their profits, then we must be careful of how we pay solvers for reporting this information, otherwise solvers may behave in a way not desired by the protocol and its users.

More specifically, we a would desire that a solver competition satisfy as many of the following properties as possible.

1. Yield the best outcome for users of the protocol.
2. Encourage solvers to disclose their best solution to the protocol.
3. Encourage solvers to disclose the costs associated with producing their solution.
4. Allow solvers to focus their resources on producing better user outcomes, rather than requiring them to optimize for intricacies of the solver competition.
5. Never require subsidization from the protocol.
6. Discourage solvers from colluding and extracting additional profits at the expense of users or the protocol.
7. Never require the solver to risk negative profit.

An auction mechanism that satisfies many of these properties is what we call a quality-adjusted 2nd price auction (QASPA).

Quality-adjusted 2nd price auction (QASPA)

Defining QASPA requires three important concepts: solution quality, solution cost, and solution utility. Solution quality is the utility ascribed by the protocol for a particular solution; solution cost is the cost associated with executing the solution on-chain; and solution utility is simply the solution quality minus the solution cost. Solvers would submit a solution to the protocol, along with the solver's best estimate of the cost of executing the solution on-chain, and the protocol would use these to compute the utility of the solution. The protocol would then select the top-utility solution's solver the auction winner, and the protocol would pay the winning solver their estimated cost to execute the solution on-chain, plus the marginal utility that the winning solver's solution provides above the next highest utility solution.

Payment to winner, given that player i wins and player j has the second best solution.

In the case where the all of the solutions' qualities are the same, this auction is equivalent to a reverse second-price auction: the solver with lowest cost solution wins, and they are paid the value of the second lowest cost solution.

Using this auction mechanism, we satisfy almost all of the properties above, with the exception that properties 5 and 7 may not hold under exceptional circumstances (see paper for more details). But analyzing auctions in theory is not enough; we also bring data to support the viability of a migration to this new mechanism.

Backtesting on historical data

Using historical data from the CoW protocol solver competition, we are able to estimate the payments that the protocol would have made if it were running QASPA on prior batches. This methodology is certainly imperfect, since solutions could change once we modify the auction mechanism; still, if we assume that the solvers were predominantly truthful and reported their best solutions historically, we might expect a similar distribution of solutions after modifying the auction mechanism, in which case we believe this methodology may be sound.

If we look at the distribution of payments made under a QASPA, we see that the significant majority of payments are below the payment made by the mean of the status quo mechanism.

[

768×544 125 KB

](https://europe1.discourse-cdn.com/business20/uploads/cow/original/1X/7d937d885cce01a2b9bf48d2c68e7b83f92339c2.png)

However, there is a fat tail of QASPA solver payments that are orders of magnitude greater than that of the status quo. These fat tail payments are so large that the mean payment from QASPA is over 2x larger than that of the status quo! This is an unacceptably large solver payment.

The way we can avoid this is by instituting a cap on the utility surplus payment. Instead of paying the winning solver their cost plus their marginal utility, we would pay them their cost plus the lesser of their marginal utility and some cap k

.

[

The formula for capped solver payments.

754×58 14.9 KB

](https://europe1.discourse-cdn.com/business20/uploads/cow/original/1X/c76d2db58a2f5717bccf7ead4027b515dd464202.png)

The formula for capped solver payments.

The formula for capped solver payments.

By instituting this cap, we set an upper bound on our incentivization of additional utility created by the top solution. When the cap is not met, the payment rule is equivalent to QASPA's; when the cap is met, the protocol limits its expenditures at the expense of encouraging better solutions. In our paper, we estimate that a utility surplus cap of 0.312 ETH

would lead to the same mean payment as the status quo solver competition, and we estimate that only 0.692% of batches would reach this cap. This means that in over 99% of the batches, we get all of the nice properties of QASPA, while in less than one percent of the batches, we have a capped utility payment. It is worth noting also that the capped utility payment is still greater than the fixed payment that solvers would receive under the status quo, and thus even when the cap is met, the payment rule provides more incentive for good solutions than the status quo.

The protocol could also save money by decreasing the cap below 0.312 ETH

. This comes with the tradeoff that more solutions will meet the cap. However, we find that this can be a reasonable tradeoff - the protocol would save 10% of its payment costs ($500k+ per year) by reducing the cap such that 1.3% of solutions meet the cap, or save over 25% of its payment costs by reducing the cap such that less than 3% of solutions meet the cap. When a solution meets the cap, this is still strictly better for solvers than the status quo today, so long as the cap is greater than the constant value paid today.

[

The protocol can save over $500k while keeping the number of affected solutions relatively low.

770×560 65.3 KB

](https://europe1.discourse-cdn.com/business20/uploads/cow/original/1X/84e7cf1d08658af91f1efc97e66f83d3170e3f0e.png)

The protocol can save over $500k while keeping the number of affected solutions relatively low.

Recommendation

We are big believers in the value of a quality-adjusted second price auction. To be clear, this auction mechanism does come with drawbacks: potential for protocol losses, occasional solution misreporting by rational solvers, and potential fragility in times of scarce block space. And there are still a host of unanswered research questions on this topic that we bring to light in our paper. Nevertheless, both theoretical and historical evidence suggest that implementing a quality-adjusted second price auction will lead to meaningful decreases in protocol spending, elimination of pennying and other negative solver behaviors, simplification of solver bidding behaviors, and, most importantly, better user outcomes.

Links

- Research paper
- Research code
- Blog post

Credit: Thomas Bosman (@tbosman) independently discovered a nearly identical quality-adjusted 2nd price auction months before I did; Marco Correia presented the idea for a similar capped payment to what is used here; and the entire CoW protocol team has supported this research.

Disclosure: This research was funded by the CoW protocol team. Neither Xenophon Labs nor any of its employees hold a financial interest in CoW protocol, COW token, nor any CoW protocol affiliates. Any opinions and results stated here are those of the authors, not of CoW protocol or any of its affiliates. This is not financial advice.