This proposal is joint work of the CoW Protocol solver team (@marco, @harisang, Alex and myself) and our friend @voyta.eth.

TL;DR

An implicit promise of CoW Protocol is that users should get an equal or better deal than if they traded against publicly available liquidity offered by prominent protocols, such as Uniswap, Balancer, or Curve. This blog post makes this promise more explicit, and suggests a simple method that solvers may implement to minimize their chances of breaking this promise.

Introduction

The objective function for solvers to maximize as defined by the current rules of the competition comprises the total user surplus. As of the date of this post, CoW Protocol settles only a minority of total volume on Ethereum, but this total user surplus does not directly consider the clearing exchange rates outside CoW Protocol. This is of particular concern in batches where user orders can be matched directly against each other within a certain price interval. In these cases, solvers currently have some freedom to choose the clearing price, and hence to determine how the achieved surplus is distributed to the users. This, in turn, may lead to solutions that can objectively be considered "unfair" with respect to the external market prices.

In our view, the uniform clearing prices (exchange rates) computed by a solver should be in line with what the users would get elsewhere. In other words, solutions should only be considered valid if they cannot be significantly improved by using publicly available liquidity.

The goal of this post is to propose a notion of "fairness" of a settlement based on the state of a set of baseline on-chain liquidity sources (AMMs). We will refer to this concept as "envy-freeness of AMMs". While it is yet to be researched how this can be most effectively added to the rules of the solver competition, we describe "capped surplus" as a simple modification of the objective function that can help solvers find "fair" solutions.

Please note that this post is about fair surplus distribution in the context of CoWs (that is, directly matching user orders on a token pair or cycle), whereas this other blog post by @harisang investigates fairness for multiple independent trading cycles of user orders and AMMs.

Envy-freeness of AMMs

For limit orders, envy-freeness mandates that all orders whose limit prices satisfy the given clearing price need to be matched. That is, "envy" refers to the quoted clearing price, but extends to the mutual relationship between orders: whenever some order is matched, all orders offering a lower ("less ambitious") limit price must also be matched.

The concept of envy-freeness can easily be adapted to AMMs: for any given clearing price, an AMM can be "envious" if it offers a better price but was not used. In the following, we will show how this can be useful for ensuring fairness for user orders.

The main idea of AMM envy-freeness is easiest to understand for two-dimensional order books. As an example, let's consider the following scenario, where the "external market price" is represented by a single AMM, with its spot price between the limit prices of two perfectly matching orders:

The picture can be read as follows:

- We're looking at a pair that involves token A (traded against some other token)

- Sell order o1 is selling A at any price above its limit price (indicated by the green arrow)

- Buy order o2 is buying A at any price below its limit price (red arrow)

- There's an AMM (whose trading curve is plotted in orange) that also sells (the upper curve) and buys A (the lower curve)

In a settlement, the AMM does not need to be used at all. Any clearing price between the two orders' limit prices yields the same amount of total user surplus and thus the same objective function value. Hence, it is currently up to the solvers to decide how the surplus is distributed. For example, setting the clearing price to the limit price of o2 would give all surplus to o1. However, this solution immediately appears "unfair" when taking the external market information (as provided by the AMM) into account. This is because o2 would have received more surplus if it had traded against the AMM. Seen from another angle, the AMM would have liked to trade at that clearing price, and thus is "envious" if it is not used. With this, it becomes obvious that the clearing price should be equal (or at least very close) to the AMM spot price: both o1 and o2 receive at least as much as if they traded against the AMM and are protected from being exploited, and the AMM itself is "envy-free".

As a slight variation of the above, consider that the order o2 is now larger than o1:

In this case, the only way to find a solution is to use the AMM, which puts the clearing price (i.e., the effective exchange rate) somewhere between the original and the new AMM spot price.

Now let's consider what could happen if the AMM spot price was lower than o1's limit price:

In this case, depending on the size and type (buy vs sell) of o1, there can be a choice between matching o2 against o1 alone, matching o2 against the AMM alone, or match o2 against both o1 and the AMM. If o2 is matched against o1 alone then the AMM would be (legitimately) envious of o1 - informally, it would have liked to trade with o2 since it could match, and even improve, the price offered by o1. To avoid making the AMM envious, the clearing price of this batch can never be higher than its new spot price, which ends up protecting order o2.

More formally, the envy-freeness constraint we would need to check reads:

$p(A) \leq new\_spot\_price(xAMM, yAMM)$ (1)

where xAMM and yAMM are the buy and sell amounts of the AMM, and p are found prices.

Asking for all solutions to be strictly envy-free could unfortunately lead to many acceptable solutions being rejected. A solver might choose to not use an AMM that is envious for legitimate reasons: because it is not cost-effective to do so, because it would imply trading AMMs against other liquidity, among others. This means that the protocol cannot require strict envy-freeness of AMMs in general, but would most likely have to work with some relaxation (whose specifics are yet to be researched). Generally, the addition of some envy-freeness condition to the rules of the solver competition that can be checked by the backend seems highly desirable, but its specifics still need further research. Meanwhile, as a simpler alternative, the following slight change to the objective function can help achieve a fairer surplus distribution in most cases.

Capped surplus

In solutions with multiple user orders, we generally want the clearing prices to reflect the current exchange rates on the external market. That is, none of the orders should receive a better-than-market price by taking advantage of another. As mentioned above, regular surplus does not take external market information into account. Alternatively, one could consider a slight modification of the objective function, where the surplus that an order can contribute to the objective function value is "capped" at the external market price.

Going back to the first example of two perfectly matching orders, where the AMM represents the state of the external market, each order would only be allowed to contribute surplus up to the AMM spot price:

[

1087×324 45.5 KB

](https://europe1.discourse-cdn.com/business20/uploads/cow/original/1X/bf79020dbe21c9a7d2c93f624a7c89437be7e18e.png)

If the clearing price equals the AMM spot price (as in the left picture), the total (capped) surplus is maximized. By contrast, setting the clearing price below the AMM spot price (as in the right picture), yields a lower total (capped) surplus value.

As another example, let's assume that order o2 requires a very ambitious limit price (requesting more than it would get on the external market):

[

1087×324 44.1 KB

](https://europe1.discourse-cdn.com/business20/uploads/cow/original/1X/a65560f37b1ae525e729fb91bbe8b6ce28c762ff.png)

Now, the capped surplus of order o2 will always be zero as its limit price is beyond the AMM spot price. Thus, only order o1 can contribute to the objective function value, which pushes the clearing price towards o2's limit price (as in the left picture), protecting o1 from being exploited.

Formally, we define capped surplus as follows:

capped_surplus := min(external_prices_based_surplus; surplus)

where:

- external_prices_based_surplus

is a theoretical surplus that would have been obtained if the trade was executed at the external exchange rate, which is

determined and provided by the backend as part of the instance. Typically, this is not attainable for a user order given that there is some price impact, AMM fees, etc.

- surplus

refers to the status-quo user surplus, based on the user's limit price and the execution price.

Summary

Both ideas that we presented help prevent a broad range of unfair solutions and surplus shifts, but in different ways: while envy-freeness of AMMs enforces fairness as a constraint, capped surplus tries to guide the solution to fair grounds via the objective function.

Currently, the solver team believes that envy-freeness of AMMs is a more elegant and long-term viable concept, although it is pending a formal specification and subsequent implementation testing.

In terms of drawbacks, we can think of the following:

Envy-freeness of AMMs

- Potentially somewhat complex implementation, resulting from technicalities such as cost of using an AMM (gas costs are not part of the above equation (1)), arbitrage, etc…

- Might be difficult to incorporate for solvers as it requires a detailed AMM modeling

- Unclear how to verify for private liquidity of solvers (i.e., liquidity sources that were not provided by the backend as part of the batch instance, but "found" and used by the solvers themselves)

Capped surplus

- Potential for surplus shifting is not fully eliminated (only soft enforcement via objective; does not necessarily guarantee fairness for each order individually)

- Heavily relies on the accuracy of the external market prices that are used (surplus shifting potential remains in cases where the given external prices are not consistent with on-chain liquidity)

- Centralization concerns - the external price is an essential input provided in a centralized way, while long-term CoW Protocol would like to abandon such centralizing elements

We are looking forward to feedback!