An Initial Approach to Order Flow Auction Design

This work was done from March-May 2022 under FRP-20

Introduction

As previously introduced, designing good mechanisms for order flow auctions will be an important aspect of decentralizing MEV. In this work, we lay out some properties we would like such a mechanism to satisfy and an auction design that achieves some of those properties under certain assumptions. Because of these strong assumptions, we don't claim that this is the mechanism that should ultimately be adopted but hope some of the design choices provide insight into the challenges of the space.

What Are Order Flow Auctions?

Summarizing some of the previous work, by order flow auctions we are referring to the mechanism by which searchers compete for users' transactions. If a searcher wins the auction they pay the user some amount and in return, they have the exclusive rights to use that transaction as part of an MEV bundle. The goal of such a mechanism is to redistribute some of the profits that previously went only to searchers and builders back to the users whose transactions are being exploited. We imagine that for any given transaction, each searcher has some value for winning the rights to that transaction based on how much MEV they can extract from it. Since upfront the user doesn't know how much MEV can be extracted from their transaction (or else they could extract that MEV themselves) an auction is run where the market value of the transaction is discovered and searchers with more value for the transaction are more likely to win.

What Properties Do We Want?

- Ease of Use for Users

The point of order flow auctions is to redistribute some of the profits searchers and builders end up extracting back to users themselves. Without order flow auctions (or an alternative mechanism) unsophisticated users might see searchers extract lots of value from their transactions. If participating in the order flow auction requires these unsophisticated users to act intelligently or is complicated to interact with, the mechanism probably won't achieve its desired goal of redistributing value back to them. Thus we seek to design mechanisms that are as easy for users to participate in as possible. Ideally, users simply submit a transaction to the mechanism and will get back some payment without them having to do anything if MEV gets extracted from their transaction; otherwise, their transaction should get executed as normal.

- Dominant Strategies for Searchers

While we can expect searchers to be more sophisticated than users, we still want to design a mechanism that provides a dominant strategy for searchers. When searchers have dominant strategies where their bid is increasing in their value for the transaction, searchers should behave in relatively predictable ways leading to efficient outcomes by the mechanism. Without dominant strategies, it is hard to analyze how the mechanism will perform with searchers behaving erratically to maximize their profits.

- Computationally Efficiency

These auctions have to take place over many transactions for each block. These auctions also have to finish fast enough so that searchers can then participate in the subsequent auction to get their bundles included by builders. With the pace at which blocks are produced this means it is crucial for these auctions to conclude very fast.

- Credible Auctions

Ideally, these auctions are credible and the platform running these auctions doesn't have to be trusted. If the auctioneer can manipulate bids then they can potentially suppress the amount of money that should be going to users or favor certain searchers in the auction. Credible auctions are designed so that any manipulation by the auctioneer can be detected by participants; forcing the platform to be accountable to following their protocol.

- Welfare Maximizing Outcomes

As is traditionally a goal in mechanism design, we want to maximize the welfare our mechanism achieves. A trivial way to satisfy the other properties would be a mechanism that ignores the bids searchers make and always allocates the transactions to no one. Clearly, this would give zero welfare and would be an extremely suboptimal solution. By aiming for mechanisms that maximize welfare constrained to the other properties, we tend to achieve an outcome that is relatively optimal. However, we emphasize here that welfare in this context is measured by the value the winning searcher has for a transaction which in turn is simply the gross profit a searcher can extract from a transaction. By virtue of the order flow auction, this should roughly correlate to the profits that individual users see as well but does not necessarily pick the outcome that would be the socially optimal outcome from everyone's perspective under the colloquial definition of social welfare.

Model

Here, we outline the formal assumptions we will be working with. One of the main challenges in designing order flow auctions is the interplay between how searchers compete in the order flow auction and the blockspace auction. A searcher only has value for a transaction they win in the order flow auction if they can win the blockspace auction. Thus for the purposes of this work, we assume that searchers have a fixed strategy for the blockspace auction and have some prior over how much they expect to pay to win that auction. We implicitly are ignoring any concerns around dominant strategies, efficiency, credibility, etc around the blockspace auction for now but informally touch on these concerns later. We also assume that a searcher only participates in the block space auction if they win the order flow auction. Weakening these assumptions is an open problem for future work.

- There is a single transaction x

being auctioned off between m

searchers

- x_i^1=1

if searcher i

wins the order flow auction and x_i^1=0

otherwise. x_i^2=1

if searcher i

wins the block space auction and x_i^2=0

otherwise. If x_i^1=0

then x_i^2=0

.

- Each searcher pays p_i^1

in the order flow auction and p_i^2

in the blockspace auction. If x_i^1=0

then p_i^1 = p_i^2=0

. If x_i^1=1

but x_i^2=0

then p_i^2=0

but i

still pays p_i^1

.

- Each searcher i

has some fixed value v_i

for winning the transaction rights

- Given that searcher i

wins the order flow auction, i

has a fixed strategy \pi_i

for participating in the blockspace auction where \pi_i

has a fixed chance Pr_{\pi_i}[x_i^2=1]=\alpha_i

of winning the blockspace auction with expected payment E_{\pi_i}[p_i^2|x_i^2=1]=\hat{p}_i

. * If extracting the MEV is not block sensitive then \pi_i

might even span over attempting to win the blockspace auction in multiple blocks where the searcher is seeking to win only once

- If extracting the MEV is not block sensitive then \pi_i

might even span over attempting to win the blockspace auction in multiple blocks where the searcher is seeking to win only once

- Searchers have quasilinear utilities where u_i(x)=x_i^2(v_i-p_i^2)-x_i^1p_i^1

Proposed Order Flow Auction

We propose running a second price auction where the winning searcher gets sole rights to use the transaction in the blockspace auction and pays the second highest bid to the user who created the transaction. The exact details of how the transaction rights are enforced aren't considered here. We can assume some sort of cryptographic protocol where the signature for a transaction is withheld until the order flow auction concludes and then is automatically privately given to the winning searcher in exchange for payment. If no searchers bid on the auction, the transaction is simply forwarded to flashbots protect and treated as a normal transaction. In this work, we only discuss the vanilla non-credible second price auction but note there are many works such as [1] that address ways of implementing second price auctions in a credible way. The fact that it is forwarded to flashbots protect is important so that searchers don't have the option of not bidding on a transaction in the hope that that transaction will be forwarded to the public mempool where they can include it in a bundle for free.

We show that a searcher i

has a dominant strategy in this auction for transaction x

. While we only describe the strategy for a single transaction, it is still a dominant strategy for i

to duplicate this strategy across the order flow auctions for each submitted transaction as long as we assume that all searchers have additive valuations across different transactions and these transactions don't interfere with each other being included in a final block.

Theorem

: If i

follows strategy \pi_i

in the blockspace auction, then it's a dominant strategy for i

to bid \alpha_i(v_i-\hat{p}_i)

in the second-price order flow auction.

Proof

:

Given all the searchers bids b_1,...,b_m

, let b_i'=\text{argmax}_{j\neq i} {b_j}

. Thus from i

's perspective, b_i'

is the highest bid apart from i

's bid.

We consider 2 possible scenarios.

Case 1: b_i'>\alpha_i(v_i-\hat{p}_i)

. In this case, i

only wins if they also bid above \alpha_i(v_i-\hat{p}_i)

. If they win then they must pay at least p_i^1=b_i'> \alpha_i(v_i-\hat{p}_i)

. Hence their total expected utility after participating in the order flow auction is E[u] = \alpha_i(v_i-\hat{p}_i) - p_i^1< 0

. If they lose they would’ve gotten 0 utility. Thus they were better off bidding \alpha_i(v_i-\hat{p}_i)

anyways where they always get 0 profit in this case and never lose money.

Case 2: b_i'\le \alpha_i(v_i-\hat{p}_i)

. Now if i

bids \beta < \alpha_i(v_i-\hat{p}_i)

, if \beta < b_i'

then i

loses and gets 0 profit. If \beta \ge b'

then i

pays p_i^1=b_i'

and gets expected profit E[u] = \alpha_i(v_i-\hat{p}_i) -b_i'

. Instead if i

bids \alpha_i(v_i-\hat{p}_i)

then i

always wins up to tie-breaking and gets profit E[u] = \alpha_i(v_i-\hat{p}_i) - b'

anyways. Thus i

is always weakly better off bidding \alpha_i(v_i-\hat{p}_i)

.

Thus in both cases, i

is at least weakly better off bidding \alpha_i(v_i-\hat{p}_i)

in the order flow auction regardless of how the other searchers bid. \Box

As a concrete example of a \pi_i

, consider a scenario where i

knows the CDF F(b)

of the distribution of the minimum price they would have to pay to have their MEV bundle including x

win the blockspace auction. Assume that the minimum price to win a blockspace auction is drawn i.i.d from this distribution every block and bidding above this price is guaranteed to win you the auction. Furthermore, assume this bundle will return the same revenue v_i

as long as it is included in any of the next n

blocks. Let b_i^*

be the solution to b = v_i - \frac{1-(1-F(b))^n}{nf(b)(1-F(b))^{n-1}}

.

Lemma

:If i

's strategy is to bid the same value in each of the next n

blocks, then bidding \max{b_i^*,0}

in each round is optimal.

Proof

: Assume that i

pays p_i^1

in the order flow auction to win the rights for x

. Then i

's expected profit from bidding b

is E[u_i^{b}] =(1-(1-F(b))^n)(v_i-b)-p_i^1

. Maximizing this by taking the derivative with respect to b

we get

\frac{\partial E[u_i^{b}] }{\partial b} = nf(b)(1-F(b))^{n-1}(v_i-b) - (1-(1-F(b))^n)= 0 \ \implies b = v_i - \frac{1-(1-F(b))^n}{nf(b)(1-F(b))^{n-1}}

Hence the b^*

that solves this equation maximizes the expected profit with this strategy. If b_i^*\le 0

then i

doesn't have an optimal strategy to participate in the block space auction following this strategy and should just bid 0. \square

Thus if i

is following this strategy, they should bid \max{(1-(1-F(b_i^*))^n)(v_i-b_i^*),0}

in the orderflow auction and then bid b^*

in subsequent blockspace auctions until they either win or n

blocks have elapsed.

Evaluating Goals

- Ease of Use

: In this mechanism, the user's experience is largely unchanged from now. The user merely submits their order to the mechanism and will receive a payment if their transaction is picked up otherwise their transaction is executed normally. The only friction is in the case where a searcher wins the order flow auction but loses the blockspace auction and the user's transaction never gets executed. In this case, the user might need to submit their transaction again after some period of time, but we don't expand on the exact details here.

- Dominant Strategies for Searchers

: The theorem above shows that if the blockspace auction is predictable then the order flow auction as described will have a dominant strategy for searchers

- Computational Efficiency

: Running a non-credible second price auction is efficient. In general, if you want credibility then the efficiency of this mechanism falls back on the computational efficiencies of extant proposed credible second price auctions.

- Trustless Auctioneers

: As in the above point, by default, this auction is not credible and requires a trusted auctioneer. However, credible second price auctions can be implemented. The credibility of the block builder is also relevant but not addressed by this work.

- Welfare Maximizing

: The welfare efficiency of this mechanism depends on what strategies different searchers take in the block space mechanism. If all the m

searchers participating in the order flow auction have the same strategy for the blockspace auction, and the other bidders participating in the blockspace auction behave identically regardless of who wins the order flow auction, we have that bids according to the dominant strategy for the order flow auction are monotonically increasing in v_i

. Thus the searcher with the highest value for a given transaction will win maximizing welfare.

Predictable Blockspace Auctions

A key aspect of the previous analysis is that the blockspace auctions are predictable in how much searchers have to pay to win given that they win the order flow auction. In general, this is important for any design that keeps the order flow auction and blockspace auction separate. By participating in the order flow auction, a searcher is inherently taking on risk where they have to pay for the transaction before they are sure that they can get the transaction included in a final block. Thus if the blockspace auction has very volatile outcomes on how much searchers might have to pay to win, then searchers won't have clear strategies on how to participate in the order flow auctions. If the searchers spend too much money in the order flow auction then they won't have enough to win in the blockspace auction. Thus risk-averse searchers might underbid their values for the order flow auctions to guarantee profits after the blockspace auction concludes, reducing how much value flows back to users.

The upside is that with the introduction of order flow auctions, the nature of competition in blockspace auctions is changed. Now searchers are only competing for blockspace instead of competing for blockspace and transaction rights simultaneously. Using a first price auction for transaction rights is what caused searchers to drive up their bids in the blockspace auction, but now that that competition happens in the order flow auction, there is no need to still use the first price auction in the blockspace phase. One mechanism that can take advantage of this would be switching the blockspace auction to an EIP-1559 style mechanism by builders where there is a predictable dynamic base fee to be included in a given block.

The idea is that blockspace becomes somewhat homogenous and should have a relatively stable level of demand across short time periods given that there is no more competition for transaction rights. Under normal circumstances, there shouldn't be large shifts in the total volume of MEV (here volume refers to the total gas usage the MEV bundles take. There might be volatility in the value of MEV over time e.g. when large trades occur but this volatility should be handled by the order flow auction and shouldn't necessarily spread to the block space auction). During periods where the volume of the MEV is shifting (maybe during an NFT mint) this mechanism reverts back to a first price type of mechanism as before and the optimal strategies for searchers in the order flow auctions become unclear, but once blockspace demand stabilizes, the guarantees of the order flow auctions should return.

The above is not a formal analysis, but merely a suggestion of the type of mechanism that could be used at the blockspace level to make the order flow auctions function efficiently. One change from the EIP-1559 mechanism that would probably be needed is for builders to not burn all of the base fees to remain competitive with other builders in the proposer-builder auction who aren't following this mechanism. In general, we don't consider how the decreased revenues builders in the presence of order flow auctions face might affect the dynamics in the proposer builder market and leave this open to future work.

Conclusion and Future Work

This work lays out some of the goals order flow auctions seek to accomplish and under some strong assumptions gives a way to implement order flow auctions and blockspace auctions that achieves most of the goals. If searchers have independent values for different transactions and the blockspace auction is relatively predictable, then a second price auction seems promising on the order flow level. Relaxing these assumptions and retaining the ideal properties are key open questions for future work. Additionally, while this work considers some of the dynamics between how order flow auctions and blockspace auctions interplay, it doesn't consider how these auctions affect the proposer-builder dynamic at all. Considering how all three of the auctions compose simultaneously and the effects on properties we want out of the proposer builder market is another important question to work out before fully implementing order flow auctions.