Decentralized Markets
Evaluating Uniform and Non-Uniform Prices

Jason Vranek
jvranek@ucsc.edu

Yilin Li
yli568@ucsc.edu

ABSTRACT

We demonstrate that having a uniform clearing price provides investors with protection from front-running, trade-collisions, cancel-collisions, and stale orders when compared to Continuous Double Auctions (CDA) in a blockchain environment. This is valuable to investors and would prevent market failures. We present a simulation environment to measure the relative performance of investors, market makers, and miners in CDA, Frequent Batch Auctions (FBA), and Kyle-Lee Flow Markets (KLF). We provide experimental results indicating that Continuous Scaled Limit Orders in a batch auction format provide the least cost to social welfare in an environment where miners choose to front-run.

1. INTRODUCTION

Current fully on-chain decentralized exchange (DEX) implementations suffer from combinations of slow trading experiences, front-running, and trade and cancel collisions, that will only scale as trade volume increases. These problems arise because the state of the exchange is dependant on the ordering of message arrivals. For example, in a volatile market, fast cancellations are necessary as an arbitrageur can snipe stale orders if they have a speed advantage. Similarly, in a fully on-chain DEX, the speeds of cancellations are dictated by gas prices, which can lead to arms races in fees. Exchanges circumvent this message ordering problem by adding centralization in the form of TECs [1] or the StarkDEX service [2]. Adding centralization may eliminate collisions and provide traders with the means to quickly change their orders off-chain, but comes at the cost of adding trust. This means trusting that the exchange does not censor or front-run which are difficult to detect. Centralization introduces a single point of failure, and a clear target to be hacked. One thing to note however is that this is still preferable to a CEX where the users must trust the exchange with custody of their assets. We propose that a uniform clearing price can mitigate the impact of front-running.

Third International Symposium on Foundations and Applications of Blockchain (FAB '20) May 1, 2020, Santa Cruz, California, USA.

2. ORDER TYPES

2.1 Standard Limit Orders (SLO)

The standard limit orders are defined as messages with three parameters: a buy-sell indicator, a quantity $Q_{\max}$, and a limit price P. A standard buy limit order conveys the message Buy up to $Q_{\max}$ (shares) at a price of P (dollars per share) or better. For an order placed at time t_0, let $Q(t_0, t)$ denote the cumulative quantity executed during the time interval $[t_0, t]$ for $t > t_0$. Let $p(t)$ denote the most recent transaction price, and define $p_{\min}(t_0, t)$ as the minimum price $p(t)$ during the interval $t \in [t_0, t]$. Then $Q(t_0, t)$ satisfies

$$Q(t_0,t) = \begin{cases} Q_{\max} & \text{if } p_{\min}(t_0,t) < P, \\ \alpha(t_0,t) Q_{\max} & \text{if } p_{\min}(t_0,t) = P, \\ 0 & \text{if } p_{\min}(t_0,t) > P. \end{cases}$$

where $\alpha(t_0, t) \in [0, 1]$.

Similarly, for a limit sell order, $Q(t_0, t)$ satisfies

$$Q(t_0,t) = \begin{cases} Q_{\max} & \text{if } p_{\min}(t_0,t) > P, \\ \alpha(t_0,t) Q_{\max} & \text{if } p_{\min}(t_0,t) = P, \\ 0 & \text{if } p_{\min}(t_0,t) < P. \end{cases}$$

where $\alpha(t_0, t) \in [0, 1]$.

If the minimum market price $p_{\min}(t_0, t)$ is above (below) the limit price P, nothing is bought (sold); if it is below (above) the limit price, the order is fully executed ($Q(t_0, t) = Q_{\max)}$); if it exactly equals the limit price, then the quantity executed $Q(t_0, t)$ depends on the rule for allocating traded quantities when it may not be possible to satisfy all demands (supplies). This allocation rule determines $\alpha(t_0, t)$, which is a monotonically non-decreasing step function of time measuring the fraction of the order executed up to time t.

With standard limit orders, the limit price P is an integer multiple of the minimum tick size, and the quantity $Q_{\max}$ is an integer multiple of the minimum lot size. The discrete price grid makes an allocation rule to determine $\alpha(t_0, t)$ necessary in that the market demand and supply schedules define quantities as discontinuous step functions of price and there may not be a unique point of intersection.

2.2 Continuous Scaled Limit Orders (CSLO)

The continuous scaled limit orders are defined as messages with five parameters: a buy-sell indicator, a quantity $Q_{\max}$, two price levels P_L and P_H (with $P_L < P_H$), and the maximum trading speed $U_{\max}$. Such an order conveys the message: Buy up to a cumulative total of $Q_{\max}$ (shares) at

maximum rate $U_{\max}$ (shares per hour) at prices between P_L and P_H (dollars per share). The trading speed or flow demand $U(p(t))$ is a function of the market clearing price $p(t)$ given by

$$U(p(t)) = \begin{cases} U_{max} & \text{if } p(t) < P_L \\ \frac{P_H - p(t)}{P_H - P_L} U_{max} & \text{if } P_L \leq p(t) \leq P_H \\ 0 & \text{if } p(t) > P_H \end{cases}$$

Similarly, the flow supply $U(p(t))$ is also a function of the market clearing price $p(t)$ given by

$$U(p(t)) = \begin{cases} U_{max} & \text{if } p(t) > P_H \\ \frac{p(t) - P_L}{P_H - P_L} U_{max} & \text{if } P_L \leq p(t) \leq P_H \\ 0 & \text{if } p(t) < P_L \end{cases}$$

If the price is strictly below (above) P_L (P_H), the trader buys (sells) at the rate $U_{\max}$. If the price is strictly above (below) P_H (P_L), the trader trades zero shares. If the price is between P_L and P_H, the demand (supply) schedule is interpolated linearly, making its slope $U_{\max}/(P_H - P_L)$. The cumulative quantity executed by time t is

$$Q(t_0, t) = \int_{t_0}^{t_0 + t} U(p(\tau)) d\tau$$

Suppose the aggregate demand and supply schedules intersect at a point where either of the two is not flat. Then the excess demand schedule $D(p) - S(p)$ is strictly decreasing in the neighborhood of the intersection. The exists a best bid price P_B and best ask price P_A, both on the tick grid, where P_A is one tick size large than P_B, and there is excess demand at the best bid and excess supply at the best ask price given by

$$D(P_B) - S(P_B) > 0 \quad \text{and} \quad S(P_A) - D(P_A) > 0$$

Define the relative order imbalance $\omega \in [0, 1]$ by

$$\omega = \frac{D(P_B) - S(P_B)}{D(P_B) - S(P_B) + S(P_A) - D(P_A)}$$

Then the market clearing price $p(t)$ is uniquely defined by

$$p(t) = P_B + \omega(P_A - P_B)$$

Intuitively, the price is a weighted average of the two prices P_B and P_A, with weights $1 - \omega$ and ω proportional to the excess demand and supply at these prices.

3. MARKET TYPES

3.1 Continuous Double Auction (CDA)

Continuous Double Auctions (CDA) are one of the most common forms of marketplaces and has emerged as the dominant financial institution for trading securities and financial instruments. Indeed, the major exchanges, like the NASDAQ and the NYSE and the major foreign exchange (FX), apply variants of the CDA model. In addition, CDA also provides a dynamic and efficient approach to the decentralized allocation of scarce resources.

In the CDA model, bids and asks may be submitted at anytime during the trading period. If at anytime, there are open bids and asks that match or are compatible in terms of prices and requirements, a transaction is executed immediately. In this auction, orders are ranked from the highest to the lowest to generate demand and supply profiles. From the profiles, the maximum quantity exchanges can be determined by matching asks with demand bids.

3.2 Frequent Batch Auction (FBA)

Frequent batch auction is proposed to eliminate the race of latency arbitrage from continuous time auctions, and by having batch auctions with some high frequency, say once every 100 milliseconds, the market would establish an equilibrium price that matches supply and demand, minimizing the proportion of unfilled buy-and-sell orders. FBAs use a standard limit order book, and FBAs do not eliminate the need for an allocation rule like time priority or pro rata order matching. Since the market clears in increments of blocks, there will typically be excess supply and demand at the market clearing price. FBAs require an allocation rule for determining which executable orders do not get executed at the market clearing price. One possibility is to give gas fee priority to orders received with higher gas fees while using pro rata matching for orders received in the same batch. Another possibility is to give time priority to orders received in earlier batches.

4. SIMULATION

4.1 Blockchain Environment

In order to evaluate the different order types in a decentralized setting, we simulate a blockchain environment with some simplifications. We assume constant block publication intervals with unlimited block sizes, which allows for all pending orders in the mem-pool to be added to the next block. A publicly observable pool of pending transactions called the mem-pool receives orders in continuous time.

For simplification, there is only a single miner in the simulation. Their role is to form a frame consisting of pending orders in the mem-pool, and publish it as a new block at a fixed interval. In order to capture the latency from propagation delays in a peer-to-peer network, the miner will wait a random delay after a block publication to form their next frame. This also allows for the existence of a single miner to capture the behavior of there being multiple miners. Upon publication, all orders are processed serially in descending order based on the orders' gas fees.

When an order is processed, the state of the public order books are changed to include the new order. How this state is changed depends on the market type being simulated. The state of every past order book is observable as in a real distributed ledger. We assume that the miner will allocate orders from the mem-pool to their frame based on the orders' gas fees.

4.2 Player Behavior

4.2.1 Investors

At times determined by a normal distribution, a random investor is chosen to generate a bid or ask order and send it to the mem-pool to be processed. There are two fixed means μ_1 and μ_2 that are respectively the centers of the bid and the ask normal distributions that the investor will price their orders from. The average of the overlap of the two distributions is the unobservable fundamental value of the asset being traded. The investor's gas fee is drawn from

a uniform distribution between [0 : 1] and trade volume is fixed at 1.0 units.

4.2.2 Miner

To simulate propagation delay in a peer-to-peer network, the sole miner waits a random delay the last block is published to construct a frame. A frame is a collection of pending orders from the mem-pool that are sorted in decreasing order based on gas fees. The miner's frame formation strategy is to select in decreasing order the N pending transactions with the highest fees.

A front-running miner is the worst-case scenario for a decentralized exchange as they can always execute their order first within a block. Thus as a simplification, the miner is only front-runner in this simulation. During the frame formation, with some probability they will duplicate a profitable order and place it at the top of their frame to be processed first with no gas fee.

A frame is kept unobservable and immutable to all players until the fixed publication time arrives. Once published it is referred to as a block. The orders are processed serially in decreasing gas cost order and according to the current simulated market's rules thereby updating the state of the public order book.

4.2.3 Makers

Each maker is randomly assigned a behavioral type that dictates their strategy and an initial random inventory. Makers observe the order books and update their inferred fundamental value by averaging across the previously seen bids and asks that have arrived to the mem-pool. This is a simplification as the true fundamental value is fixed in this simulation.

Each maker will have an assigned type of either Aggressive, RiskAverse, or Random. These types are used to determine how the maker chooses to price their orders, their gas fee, and set their order's trade volume. Aggressive makers have tighter spreads, RiskAverse makers have wider spreads, and Random makers are uniformly distributed between the two. Each maker's pricing strategy is a function of their inventory with the goal to operate with minimum held inventory. Aggressive makers will place strictly higher gas fees than the mean, RiskAverse at the mean, and Random normal around the mean.

To incentivize makers to reach zero inventory, a tax is imposed on makers after every batch. Every unit of inventory, whether positive or negative, is taxed by a configurable amount. At the end of the simulation, liquidation occurs, when all the makers with negative inventory must purchase that many shares at the fundamental value, and all the makers with positive inventory must sell at the fundamental value.

The flow of a maker can be summarized as follows: Makers will track all of the orders submitted to the publicly observable mem-pool and infer a fundamental price. The maker's behavioral type determines their spread, and their current inventory determines how to shift their spread relative to the inferred fundamental price, as well as how to set their trade volumes. Gas prices are determined based on their behavioral type. Once a bid and ask pair has been created, they are sent to the mem-pool to be mined.

4.3 Simulated Markets

We simulate three different market formats: CDA, FBA, and a variation of FBA using CSLO's which we will denote as a Kyle-Lee Flow market (KLF). The following overview of the simulation is equivalent across the market formats with the differences being in the order types and the way orders are processed within a block.

Investors continuously enter orders priced by normal bids and asks distributions at random times. (SLO's for CDA and FBA, CSLO's for KLF). After a block is published, Makers infer fundamental price from previously seen mempool orders and produce bid and ask orders based on their behavioral type. All orders are added to the mem-pool in continuous time.

The miner waits for the propagation delay then forms a frame from the current N mem-pool orders, inserting their front-run order with some probability. The Miner will now sleep (mine) until publication time, and during this time orders may still arrive to the mem-pool but will not be added to the frame. At publication time, the orders are processed according to the market type begin simulated:

- **CDA**: For each SLO in the frame, if it is a bid, then we check it against the best ask. If $bid_{price} \geq bestask_{price}$ they "crossed" and transact at $bestask_{price}$, otherwise it will be added to the bids book. Likewise for an ask if $ask_{price} \leq bestbid_{price}$ they transact at $bestbid_{price}$, otherwise it will be added to the ask book. New orders can potentially fill multiple orders if they continue to cross the next best order in the book.
- **FBA**: SLOs in the frame are added to the order books and sorted by price. A batch auction calculates a uniform clearing price based on all the current bids and asks. As a simplification the auction occurs at the end of the block after the last order has been added to the order book. Inventory is exchanged at the uniform clearing price in price-priority order.
- **KLF**: CSLOs in the frame are added to the order book and sorted by price. The uniform clearing price can be found via a binary-search through the aggregate supply and demand schedules for a crossing point. As a simplification the clearing price search occurs at the end of the block after the last order has added to the order book. Each order transacts at the clearing price with $Volume = Schedule(Clearing\ Price)$.

The leftover orders remain in the order book for the next batch and the miner waits until the propagation delay expires to form the next block and this process repeats.

4.4 Performance Metrics

4.4.1 Volatility of Price

We measure the price volatility as the standard deviation of price changes under different market formats.

$$Std(P_t - P_{t-1})$$

4.4.2 Pricing Deviations from Fund. Values

We measure the pricing deviations from fundamental values as the root mean squared deviation (RMSD) of the transaction prices relative to the fundamental value.

$$RMSD = \sqrt{\frac{1}{t}\int_0^t (P_\tau - V_\tau)^2 d\tau}$$

4.4.3 Social Welfare Cost

We measure the social welfare cost as the sum of gas fees collected by the miner, the tax on inventories paid by makers and the other profits of miner/makers. Thus, it becomes the difference between total profits of all players and the investors' profits.

4.5 Results

4.5.1 Varying Maker Spread

Table 1: Summary Statistics for Simulation Data

(Varying Maker Spread)

	KLF		FBA		CDA	
Liquidation	no	yes	no	yes	no	yes
Panel.A (maker spread = 1.0)						
$Std(P_t - P_{t-1})$	0.06	0.06	0.20	0.20	0.25	0.25
$RMSD(P_t - V_t)$	0.12	0.12	0.81	0.81	0.79	0.79
Welfare Cost	220.44	235.88	4016.91	117.50	4304.02	218.76
Panel.B (maker spread = 1.5)						
$Std(P_t - P_{t-1})$	0.07	0.07	0.28	0.28	0.32	0.32
$RMSD(P_t - V_t)$	0.14	0.14	0.99	0.99	0.99	0.99
Welfare Cost	227.56	311.75	5874.10	117.05	6604.63	158.01
Panel.C (maker spread = 2.0)						
$Std(P_t - P_{t-1})$	0.07	0.07	0.32	0.32	0.38	0.38
$RMSD(P_t - V_t)$	0.12	0.12	1.13	1.13	1.10	1.10
Welfare Cost	296.70	344.66	7406.14	133.39	8400.35	159.60
Panel.D (maker spread = 2.5)						
$Std(P_t - P_{t-1})$	0.07	0.07	0.39	0.39	0.43	0.43
$RMSD(P_t - V_t)$	0.13	0.13	1.26	1.26	1.20	1.20
Welfare Cost	346.90	359.56	8912.75	89.52	9239.49	114.26
Panel.E (maker spread = 3.0)						
$Std(P_t - P_{t-1})$	0.07	0.07	0.44	0.44	0.47	0.47
$RMSD(P_t - V_t)$	0.15	0.15	1.30	1.30	1.33	1.33
Welfare Cost	250.68	398.13	9899.88	125.58	10776.88	124.95
Panel.F (maker spread = 3.5)						
$Std(P_t - P_{t-1})$	0.08	0.08	0.48	0.48	0.51	0.51
$RMSD(P_t - V_t)$	0.15	0.15	1.45	1.45	1.40	1.40
Welfare Cost	438.28	444.60	10866.72	150.55	11289.83	172.35
Panel.G (maker spread = 4.0)						
$Std(P_t - P_{t-1})$	0.07	0.07	0.52	0.52	0.53	0.53
$RMSD(P_t - V_t)$	0.18	0.18	1.48	1.48	1.40	1.40
Welfare Cost	256.58	457.52	11648.46	192.23	12065.77	226.35

Table 1 reports the performance metrics when varying the market makers' spread. From Panel A to Panel G, the maker's base is increased from 1.0 to 4.0 in increments of 0.5.

We have the consistent result that KLF using CSLOs gives the least price volatility, the least pricing deviation from fundamental value, and the least welfare cost compared to FBA and CDA. By comparison, CDA has the worst performance, partly because there is a time priority involved. One point worth the noting is the social welfare cost. For CSLOs, the welfare cost does not differ a lot whether taking the liquidation into account or not and it is relatively lower when there is no liquidation. On the contrary, for FBA or CDA, the social welfare is very positively affected when the liquidation is taken into consideration. This huge difference is largely due to the cut in makers' profits and the raise in investors' profits when liquidation is mandated as is shown in the raw data.

From the comparison among panels, we can see when makers are least aggressive (with the greatest maker spread), there is the least pricing deviation from the fundamental value, although the price volatility varies.

4.5.2 Varying Miner Front-Run Probability

Table 2 reports the performance metrics when varying the miner's probability of front-running. From Panel A to Panel E, the miner's front-running probability increases from 0 to 1 in increments of 0.25.

From each panel, KLF always results in the least price volatility, the least pricing deviation from the fundamental value, and the least social welfare cost. Additionally, CDA has a quarter more of the volatility of price and pricing deviation than the FBA. The liquidation only has an impact on the social welfare, which results in a greater social welfare cost when taken into account for flow orders whereas a much lower welfare cost for FBA and CDA. The similar reasons as the above analysis also apply here. As we increase miner's probability of front-running, we observe an increasing trend in price volatility and RMSD.

Table 2: Summary Statistics for Simulation Data

(Varying Miner Front-Run Probability)

	KLF		FBA		CDA	
Liquidation	no	yes	no	yes	no	yes
Panel.A (front-run prob. = 0.00)						
$Std(P_t - P_{t-1})$	0.06	0.06	0.16	0.16	0.22	0.22
$RMSD(P_t - V_t)$	0.15	0.15	0.71	0.71	0.77	0.77
Welfare Cost	49.89	241.55	3358.11	113.89	3799.57	178.21
Panel.B (front-run prob. = .25)						
$Std(P_t - P_{t-1})$	0.06	0.06	0.18	0.18	0.22	0.22
$RMSD(P_t - V_t)$	0.13	0.13	0.76	0.76	0.79	0.79
Welfare Cost	125.46	255.78	3846.34	96.53	4080.50	179.32
Panel.C (front-run prob. = .50)						
$Std(P_t - P_{t-1})$	0.06	0.06	0.19	0.19	0.23	0.23
$RMSD(P_t - V_t)$	0.12	0.12	0.73	0.73	0.74	0.74
Welfare Cost	233.78	246.06	3895.13	98.84	4227.14	163.99
Panel.D (front-run prob. = .75)						
$Std(P_t - P_{t-1})$	0.05	0.05	0.19	0.19	0.25	0.25
$RMSD(P_t - V_t)$	0.12	0.12	0.79	0.79	0.81	0.81
Welfare Cost	-3.31	250.89	3441.22	127.27	4721.92	202.19
Panel.E (front-run prob. = 1.00)						
$Std(P_t - P_{t-1})$	0.06	0.06	0.20	0.20	0.25	0.25
$RMSD(P_t - V_t)$	0.12	0.12	0.81	0.81	0.79	0.79
Welfare Cost	220.44	235.88	4016.91	117.50	4304.02	218.76

In summary, the results demonstrate that CSLOs, provide the greatest reduction in the volatility of price and the pricing deviation from the fundamental value, resulting in a more efficient market outcome.

5. CONCLUSION

We ran simulations to study and compare the performance of CDA, FBA, and KLF market formats in the blockchain environment. Each market format is studied under different conditions, varying the makers' spread and the miner's probability of front-running.

We find that KLF provides the least price volatility, the least pricing deviation from fundamental value, and the least welfare cost when compared to the CDA and FBA markets. When market makers are less aggressive submitting their bid-ask pairs, it decreases the volatility of price as well as the pricing deviations. From the miner's perspective, the greater probability of front-running results in a more volatile price and greater price deviation from the fundamental value.

Future work for this simulation can include smarter miner and maker strategies, and potentially endogenous investors. We wish to investigate how modifications to the simulation parameters and player strategies affects the market's outcome.

6. REFERENCES

[1] https://blog.0xproject.com/introducing-0x-protocol-v2-9f5bda04d38d

[2] https://www.starkdex.io

[3] Aldrich, Eric Mark and López Vargas, Kristian, Experiments in High-Frequency Trading: Comparing Two Market Institutions (February 21, 2019).

[4] Kyle, Albert (Pete) S. and Lee, Jeongmin, Toward a Fully Continuous Exchange (July 4, 2017)

[5] https://github.com/JasonVranek/MarketSim