High-Frequency Trading on Decentralized On-Chain Exchanges

Liyi Zhou *, Kaihua Qin *, Christof Ferreira Torres †, Duc V Le ‡ and Arthur Gervais *

* Imperial College London, United Kingdom Email: liyi.zhou@imperial.ac.uk, kaihua.qin@imperial.ac.uk, a.gervais@imperial.ac.uk
† University of Luxembourg, Luxembourg Email: christof.torres@uni.lu
‡ Purdue University, United States Email: le52@purdue.edu

---

**AMM DEX**
- Blockchains enable peers to transact without trusting third-party intermediaries.
- Smart contracts are programs stored on the blockchain.
- Decentralized exchange (DEXs) allow parties to participate in financial markets while retaining full custody of their funds.
- **Liquidity Provider:** a market participant that provides liquidity.
- **Liquidity Taker:** a market participant that buys or sells one asset in exchange for another asset, by taking the liquidity offered by liquidity provider.
- **Automated market maker (AMM) DEXs** algorithmically perform market making using smart contracts.

![Image](Image)

**Slippage Protection**
- There are two types of slippages:
  - Expected slippage is the expected increase or decrease in price based on the (i) pricing formula; (ii) trading volume; (iii) available liquidity.
  - **Unexpected slippage** is the additional slippage. This is typically caused by other market participants.

**Constant Product Pricing Formula**
- *Instant liquidity*
- *irrespective of the trade size*
- *Purchase of Y increases price of X and decreases the price of Y*
- *Ratio of asset X and Y sets the price*

\[ x \times y = k \]

**Slippage Protection**
- Parity prioritises local and retracted transactions first, and penalises transactions with heavy computation.
- Transaction ordering is more complicated nowadays, as miners start to provide transaction reordering as a service.

**How miners order transactions**

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Number of Blocks</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty Block</td>
<td>55,545</td>
<td>0.0234</td>
</tr>
<tr>
<td>Order per Gas Price</td>
<td>1,862,800</td>
<td>0.7653</td>
</tr>
<tr>
<td>Order per Parity Default</td>
<td>384,120</td>
<td>0.1020</td>
</tr>
<tr>
<td>Unknown Ordering</td>
<td>68,569</td>
<td>0.0203</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>2,372,084</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

**Predatory trading**
- In traditional markets, the predatory trading strategy of front-running involves exploiting non-public information about a pending trade. If the asset price is expected to rise/fall as a result of the pending trade, the front runner will seek to buy/sell the asset before the pending transaction executes.

**AMM DEXs** aim to mitigate malpractice by providing complete transparency about (i) the available liquidity for asset X and Y; (ii) all performed trades; (iii) all pending trades on the P2P network; (iv) the pricing formula.
- However, AMM DEXs also exacerbate malpractices, such as sandwich attacks.

**Sandwich Attack**
- **Liquidity Taker Attacks Liquidity Taker**
  - The victim transaction \( T_{V} \) specifies its slippage protection based on the AMM state of block \( V \).
  - The adversarial’s goal is to include \( T_{A1}, T_{V}, T_{A2} \) in the same block.
  - \( N \) is \( k \) in that exact sequence.
  - Not every victim transaction yields a profitable attack. We quantify a minimum profitable victim input, under which an adversary will be unable to make a profit.

**Sandwich Attack - Liquidity Taker Attacks Liquidity Taker**
- We present a novel sandwich attack, where a liquidity provider targets a victim liquidity taker.
- The attacker pays higher transaction fees.
- The attacker foregoes the commission fees for the victim’s transaction.

**Multiple adversaries**
- We assume all adversaries are rational and attack with the parameters defined in table below.
- Our results suggest that having multiple attackers does in expectation divide the total revenue.