The Impact of Market Conditions and Fee Algorithms on the Design of a Competitive AMM

Vladislav Volosnikov
Algebra Protocol
vv@algebra.finance

Alexander Pimenov
Algebra Protocol
ap@algebra.finance

Vladimir Tikhomirov
Algebra Protocol
vt@algebra.finance

January 29, 2024

Abstract

Automated market makers (AMM) are an important part of the DeFi space and are attracting a lot of attention from developers and researchers. There are more than 1,000 DEXes with $14B total liquidity what leads to tough competition.

In this work, we examine the impact of various external and internal parameters on the profitability and competitiveness of an AMM under ideal market conditions. We also propose a number of different approaches to managing liquidity pool fee value, including a dynamic fee mechanism. The influence of the parameters, as well as the effectiveness of the fee management algorithms, are tested using the model of ideal market. This article demonstrates the impact of market conditions, blockchain features and the fees used on the income of liquidity providers, which is one of the key metrics for DEX. Based on our research AMM developers could improve the existing solutions and design new one.

1 Introduction

Automated market makers (AMMs) currently remain one of the most important components of the DeFi sphere. Decentralized applications that enable simple and efficient exchange of digital assets are extremely popular both among end users and various protocols built on blockchains.

Traditional AMMs use liquidity to operate, which is represented in the form of reserves of corresponding tokens provided by liquidity providers. Protocols are interested in increasing the amount of liquidity, as this leads to increased efficiency of AMM, increased attractiveness for users, improved trading volume and protocol income indicators. Therefore, in the face of fierce competition between decentralized exchanges, a key factor for the success of the protocol is attracting more liquidity, which ultimately comes down to increasing the profitability of providing liquidity.

Historically, competition between AMMs has led to the creation and development of various solutions aimed at attracting more liquidity and using it more efficiently. One of the solutions was additional incentives for liquidity providers using additional rewards (farming and its variants). This approach attracts liquidity, but requires the use of protocol's or project's own funds, or the redistribution of protocol income. For this reason, additional incentives are difficult to maintain consistently over the long term, and the attracted "hired liquidity" easily flows into other protocols following new rewards after new rewards.

On the other hand, new AMM design options continued to appear with new invariants and price curves, methods for calculating the result of the exchange and user rewards, which was aimed at increasing the efficiency of using liquidity for various assets that have some previously known qualities (for example, most of the time the price remains around the peg value). A logical result of this process was the emergence of generalized AMMs with the ability to form a price curve of arbitrary shape, such as UniswapV3 [Hayden Adams et al., 2021], Algebra [Algebra Protocol, 2021] and other solutions with concentrated liquidity.

At the moment, the AMM design space continues to expand. Classic AMMs have disadvantages: divergence loss (impermanent loss) caused by any price movement, losses (lost profits) in favor of toxic flow, LVR [Jason Milionis et al., 2023], the need for a deep and serious assessment of the risks of their use. For these and other reasons, generating income from passive liquidity provision can be challenging, and the development of new mechanisms, approaches and tools in the DeFi space is aimed at solving such problems.

The creation and development of new effective variants of AMM is a key task of the Algebra team. Many publications and discussions are devoted to research in the direction of combating the negative features of AMM; this is a very broad and difficult task, the solution of which requires a lot of effort. However, for productive development in this direction, a deep understanding of the principles and features of trading on the blockchain is necessary, the identification of important parameters and conditions, detailed studying their influence on the effectiveness of AMM. This article describes the approaches we used to conduct an experimental study of the characteristics of the properties of AMM, and also proposes a number of options for additional mechanisms to increase profitability, along with the results of testing their effectiveness in the simulation system we use.

The purpose of this article is to explore various ideas and approaches to improve current AMM algorithms by controlling the value of the fee. The simulation used for this study compares classical AMM with a range of innovative proposals and different configurations to form generalized but reasonable assessments of their areas of applicability. When comparing two approaches within the same blockchain, we largely abstract from the behavior of the liquidity provider, assuming a comparison only of the relative liquidities in the pools, and the similarity of the intervals of the allocated liquidity. This simplification is possible due to the fact that in both cases, liquidity providers observe the same market and price movement.

2 Description of the simulated AMM

In the current study, we use the most common variant of the classic AMM, used in the second version of the Uniswap protocol [Hayden Adams et al., 2020]. The operating logic of such an AMM is based on ensuring the fulfillment of an invariant of the following form:

$$X * Y = K \tag{1}$$

Where X and Y are available reserves of pool tokens (token0 and token1, respectively). K - constant.

This invariant (1) is also called the constant product function, due to which the entire class of corresponding AMMs is also called CPFMM (constant product function market maker). This design of a decentralized exchange remains one of the most popular and underlies, among other things, the Algebra protocol, and therefore best suits the goals and objectives of the study.

If the invariant is preserved, the current instantaneous (marginal) price can be defined as:

$$P = \frac{Y}{X}$$

In light of the development of the concentrated liquidity approach, we move to another presentation of the same principles underlying the CPFMM:

$$\Delta\sqrt{p} = \frac{\Delta y}{L}$$
$$\Delta\frac{1}{\sqrt{p}} = \frac{\Delta x}{L}$$

Where L is liquidity constant, P is current marginal price in the liquidity pool.

Nowadays, the CL-AMM (AMM with concentrated liquidity) approach is widely used, including Algebra AMM. Liquidity in such protocols is not evenly distributed; different price ranges may use different liquidity values. To simplify the simulation, we ignore changes in liquidity when the price crosses the boundaries of the ranges and consider liquidity as a fixed parameter.

For each swap, from the amount of incoming tokens, a specified percentage is withheld as a fee for the swap on this AMM. The collected fee is distributed among liquidity providers and acts as a source of their income when using the protocol.

3 Description of the market model and simulation

We use simulation modeling of the token market. To do this, the following general assumptions and assumptions are made:

- instantaneous dissemination of information and instant decision-making by agents in the market. There are no delays in receiving and processing information about market conditions;
- equal and complete access to information available to the corresponding class of agents;
- absolute rationality of market participants. Each agent in any situation maximizes the efficiency/profitability of its actions;

The assumptions and hypotheses used are very strong: in reality they are either not fulfilled or are only partially fulfilled. For this reason, some aspects of the simulation, especially those related to user behavior, may differ very significantly from what is observed in reality. This fact will be discussed in more detail in the final sections of this article.

3.1 Description of the blockchain within the simulation

The market simulation is carried out under the given blockchain parameters. The blockchain model is represented in the form of sequential blocks of transactions formed at fixed intervals (block time). Each block corresponds to a specific point in time (timestamp).

The inclusion of each transaction in the block entails a fee - payment for the gas spent. In the current simulation, the gas price does not change over time and is a fixed parameter. The block has an unlimited size, and there is no competition for inclusion in the block.

The following blockchains were used as part of the simulation:

Blockchain	Block time, seconds	Swap cost on default AMM, USD
Ethereum	12	12
Polygon	2	0.2
Intermediate option	6	0.2

3.2 Informed flow

The fundamental behavior of a trading pair is defined by an external price source - it is assumed that there is an ideal market price at each moment in time. To bring the simulation closer to reality, second-by-second values from Binance CEX for a certain period of time are used as ideal prices. As the design of CPFMM suggests, it is unable to independently determine the current price and fully relies on market participants. Within the framework of our model, price discovery on simulated AMMs is carried out using a separate class of market agents - arbitrageurs (informed flow), who have knowledge of the current ideal price. These agents instantly execute trades on the AMM and on the external CEX.

The simulated CEX has absolute elasticity (infinite liquidity) and allows an arbitrageur to instantly execute transactions with a given fee. It is important to note that in practice there are time delays, different sources of liquidity are used for arbitrage, and CEXs have limited liquidity. However, at this stage we are neglecting such details of the real market.

Also, by default, there is no competition between arbitrageurs - within the model, this is represented by the presence of a single arbitrageur with absolute efficiency. The lack of competition ignores the impact of the auction in reality: the cost of gas or the reward for including a transaction in the right place in the block. This simplification increases the profitability of arbitrage in the model. In cases where competition among the informed flow plays an important role, this will be noted separately.

As a result, at each moment in time, the arbitrageur makes a decision about whether it is necessary to carry out arbitration, as well as about the required volume of arbitrage, based on profitability. If the income exceeds the transaction costs, arbitrage occurs. The arbitrage transaction is always simulated as the very first one in the block.

3.3 Uninformed flow

The second important component of our market model is the so-called uninformed flow, which simulates the overall behavior of random trading participants (retail, end users, other protocols, and so on). The behavior of this segment of agents differs from informed flow - in each case, the agent needs to swap a certain amount of a particular token, and the best (most effective) way to carry out such an exchange is selected.

The maximized metric of swap efficiency is the total cost of tokens received as a result of the exchange, minus the transaction fee paid (gas payment). In this case, each "uninformed" agent participating in the trade chooses among two paths:

1. Perform an entire swap on one of the AMMs;
2. Exchange some of the tokens on one AMM, and some on another. In this case, the agent pays gas for each of the swaps on the corresponding AMMs.

The choice of method for conducting a transaction follows the logic of aggregators: in steps of 10%, the search for the most optimal division of the swap along possible routes is carried out.

The decision to perform a swap by an uninformed agent is made in accordance with the observed distribution of time between swaps on the corresponding real reference liquidity pool (fig. 1). The most liquid CPFMM pool on the reference blockchain is taken as a reference.

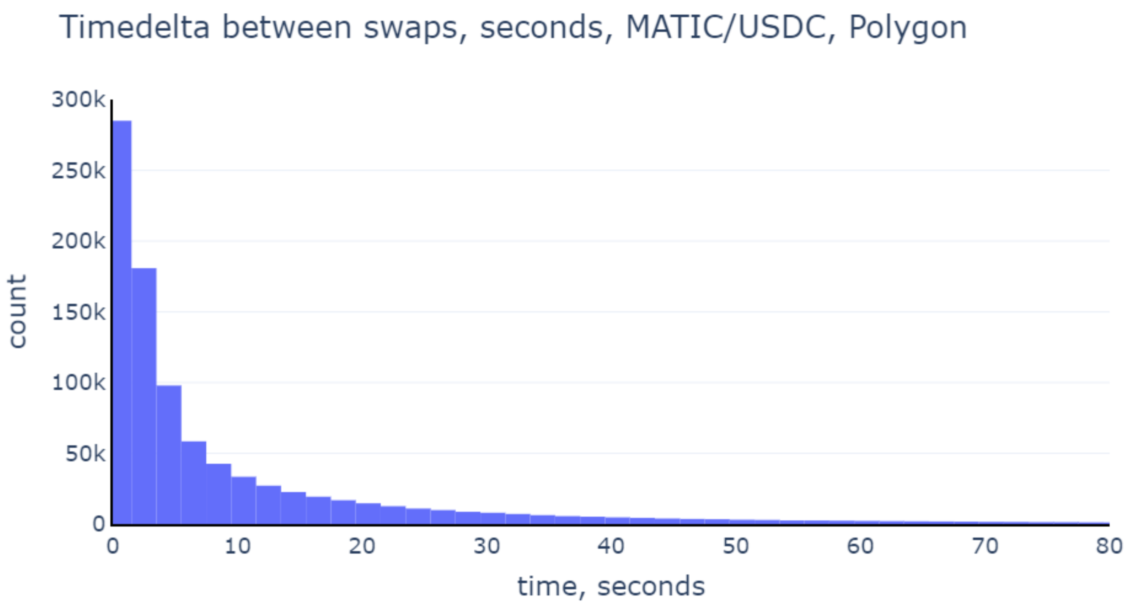


Figure 1: Timedelta distribution between swaps in MATIC/USDC pool on Polygon

The observed shape of the time distribution between swaps is similar to the shape of the Poisson distribution:

$$p(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

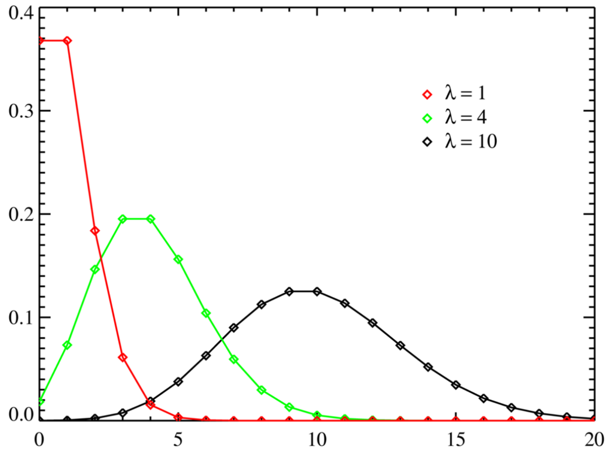


Figure 2: Examples of Poisson distributions

For this reason, the simulation uses the assumption that the corresponding distribution of the flow of applications has the form of a Poisson distribution (fig. 2) with fixed parameters within each hour of the day, and the density of their flow is taken from an analysis of the statistics of exchanges in the corresponding pair on UniswapV3.

After deciding to conduct a swap, the trader determines the type of transaction (buy/sell) and the number of tokens. In each case, the decision is made on the basis of a random distribution in the form of a Laplace distribution with given parameters:

$$f(x) = \frac{\alpha}{2} e^{-\alpha|x-\beta|}$$

The current study assumes that the Laplace distribution satisfactorily reflects the properties of uninformed flow (fig. 3).

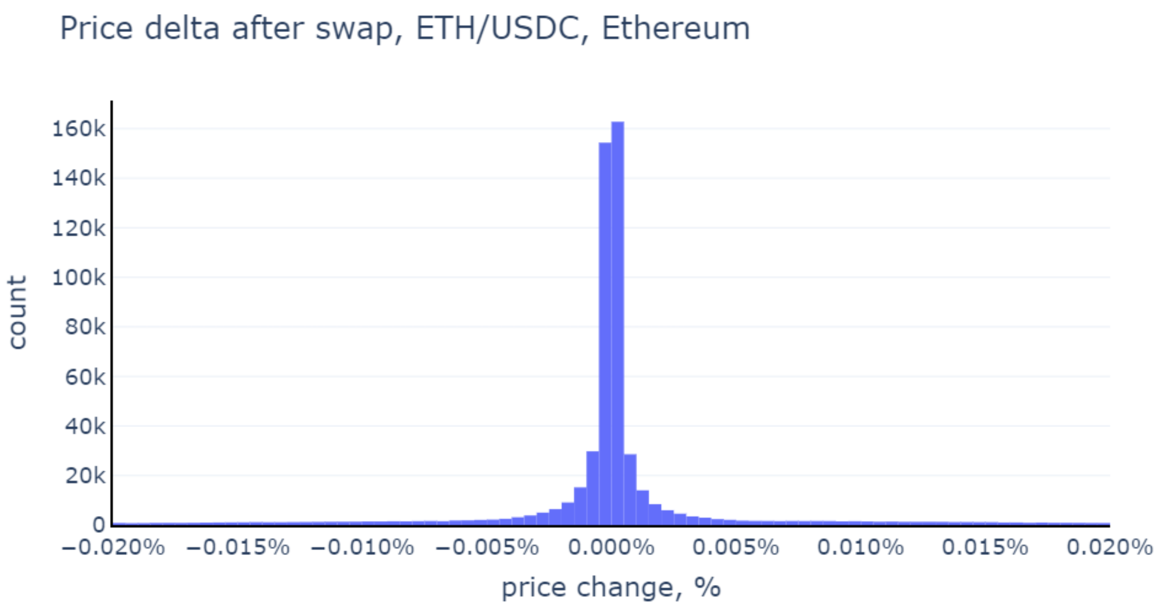


Figure 3: Distribution of price deltas after swap

It is known that market properties can change over time and also depend on a number of external parameters. One of these parameters is seasonality - trading activity depends on the moment in time. In order to increase the realism of the current study, we take into account the influence of time of day on the activity of uninformed flow. As a result, every hour according to UTC, the swap volume distribution parameters change (fig. 4).

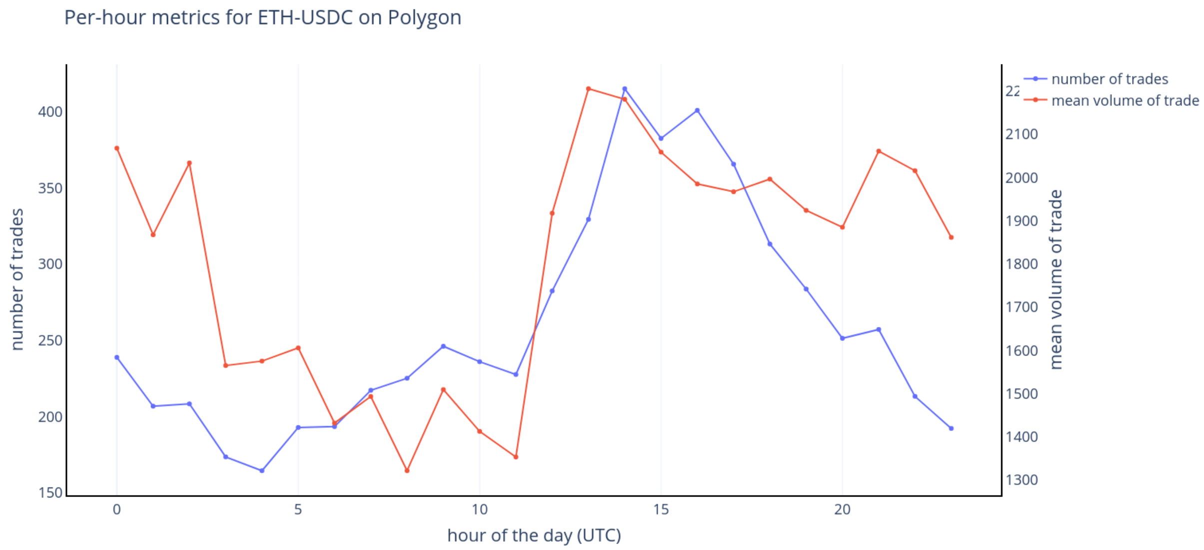


Figure 4: Per-hour metric for ETH-USDC pair in Polygon

We assume that the uninformed flow model used allows us to draw conclusions about some properties of the "ideal real market". Obviously, in reality the number of parameters and dependencies is much larger. However, at this stage we approximate the behavior of agents using the principles described above. The following effects and features are ignored, among others:

- the FOMO (agiotage) with sharp price movements
- other seasonal effects (holidays, weekends)
- the presence of other external events that change distributions and parameters

3.4 Summary description of the simulation

The components of the model described above together form the final simulation stand, each step of which consists of the following actions:

1. Determining the current point in time on the blockchain (timestamp)
2. Conducting, if necessary, arbitrage of all present AMMs with an external CEX at the current abstract ideal market price
3. Simulation of uninformed flow

Each individual experiment is described by the following parameters:

1. Blockchain used

2. Simulated token pair
3. Set of competing AMMs used

Moreover, each AMM may have different liquidity, swap cost (amount of gas), commission, and additional features.

4 Research of the influence of changing model parameters

4.1 Description of the study

In this section, on the described simulation stand, a variational study of the influence of the parameters of classical CPF-AMM is carried out. Each specific experiment involves conducting a number of simulations with different values of the parameter under study.

The comparison is made between the reference standard CPF-AMM and the modified version under study. As a result, the final statistics are compared, including the ratio of the collected fee to the liquidity value.

4.2 Variation of AMM fee value

In this experiment, each specific simulation is performed at a given swap fee in pool1. The commission for swap in pool0 is fixed and equal to 0.05%.

After simulating a highly liquid trading pair ETH/USDC on the Ethereum blockchain, the following result was obtained (fig. 5):

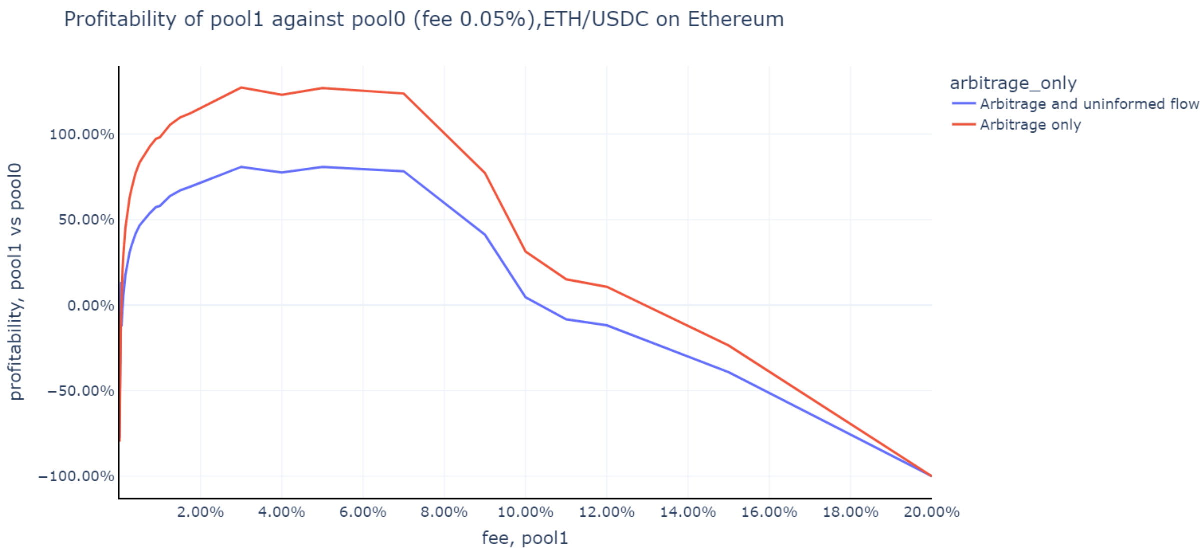


Figure 5: The result of fee variation for ETH-USDC on Ethereum

Separately, fig. 6 shows low fee values close to pool0 (0.05%).

Based on the data obtained, it follows that if there are only arbitrage transactions, the most profitable strategy is to use a high commission value. At the same time, the global

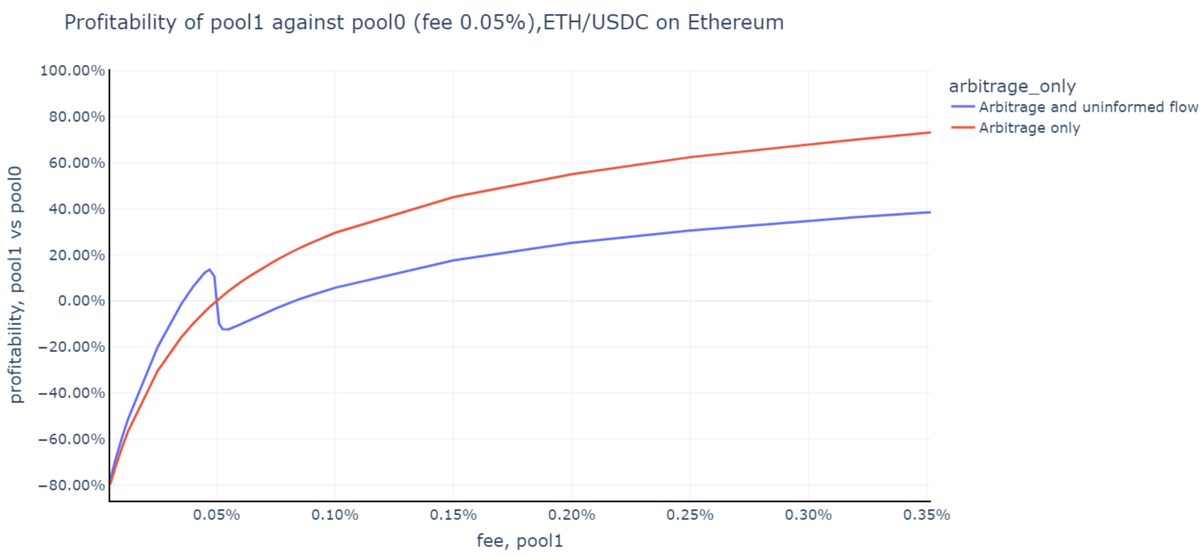


Figure 6: The result of fee variation for ETH-USDC on Ethereum, small values

maximum profitability is unique and is achieved with a fee in the region of 4-8%. Up to a certain point, increasing the commission increases the profitability of the pool, but if the values are too high, the profitability decreases rapidly: if the volatility of the trading pair is not enough, then too high a commission leads to the fact that the pool ceases to be used.

In the presence of rational uninformed flow, the situation changes significantly: from the resulting graphs it is clear that in the presence of random trading, firstly, the strategy of using extremely high commissions becomes not so profitable, and secondly, a local extremum of profitability arises when the commission decreases below the competitor. This local extreme occurs due to the fact that with lower fees in highly liquid pools, a significant share of random volume goes to the pool with a lower fee.

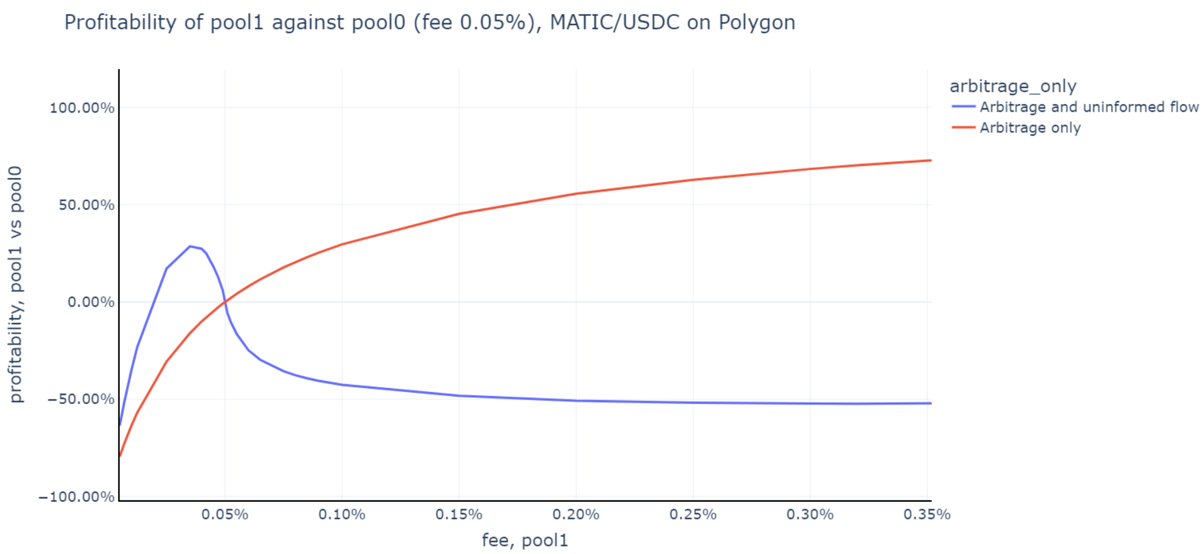


Figure 7: The result of fee variation for MATIC-USDC on Polygon, small values

After simulating a highly liquid trading pair MATIC/USDC on the Polygon blockchain, the following result was obtained (fig. 7).

Based on the results obtained, we can conclude that fee values are extremely important in competition between AMMs. With sufficient activity of rational uninformed flow, a small reduction in commission should lead to the "interception" of most uninformed trades, which will lead to increased profitability for providers on this AMM. On the other hand, setting "protectively high" commissions allows DEX to receive significant income if the traded pair has a sufficiently high price volatility.

It should be noted that with an appropriate AMM design, the fee value in the liquidity pool is easily changeable. However, in practice, making an accurate decision on the value of the commission requires a detailed analysis of other parameters of the system and the market, successively approaching the optimal value after assessing the observed changes. The Algebra protocol provides the ability to quickly and safely change the pool commission without liquidity migration.

4.3 Variation of AMM liquidity value

In this experiment, it is assumed that the simulated system has a fixed total liquidity. This approach is due to the fact that changes in total liquidity can lead to changes in the influence of random trading on the evolution of the model, which can lead to incorrect conclusions.

The simulated system consists of two AMMs: pool0 and pool1. The liquidity of pool1 is specified using the following system of formulas:

$$L_{pool0} + L_{pool1} = L_{sum}, L_{sum} = const$$
$$L_{pool1} = L_{pool0} * k, k = const$$

Each individual simulation uses a fixed value of the parameter k, which describes the proportion of liquidity in pool1 to liquidity in pool0.

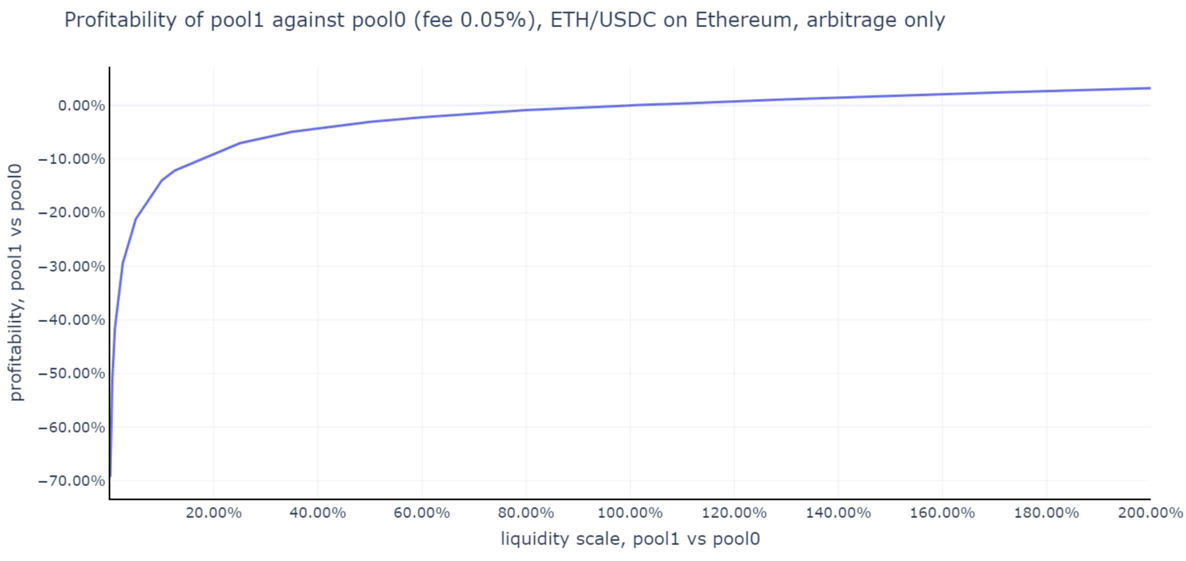


Figure 8: The result of liquidity variation for ETH-USDC pool in Ethereum, arbitrage only

In the case of using a blockchain close to Ethereum, the ETH/USDC trading pair, in the complete absence of uninformed flow (only arbitrage is simulated), the following result is obtained (fig. 8).

The profitability of liquidity provision grows with the size of the liquidity: each new liquidity provider increases the profitability for all protocol providers.

If there are rational random trades, we receive the following data (fig. 9):

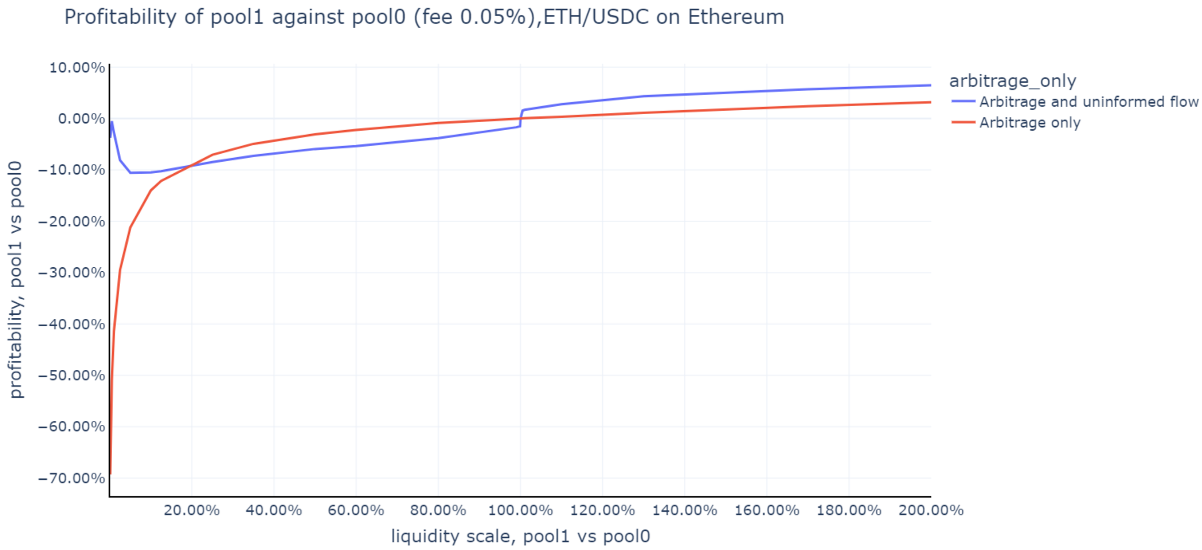


Figure 9: The result of liquidity variation for ETH-USDC pool in Ethereum

The obtained result allows us to conclude that under the described conditions, an increase in liquidity also leads to an increase in the efficiency of the entire liquidity of the pool. Thus, a more liquid pool will be more attractive, all other things being equal. However, all other things being equal, small differences in liquidity in highly liquid pools are not such a significant factor in the presence of random trading.

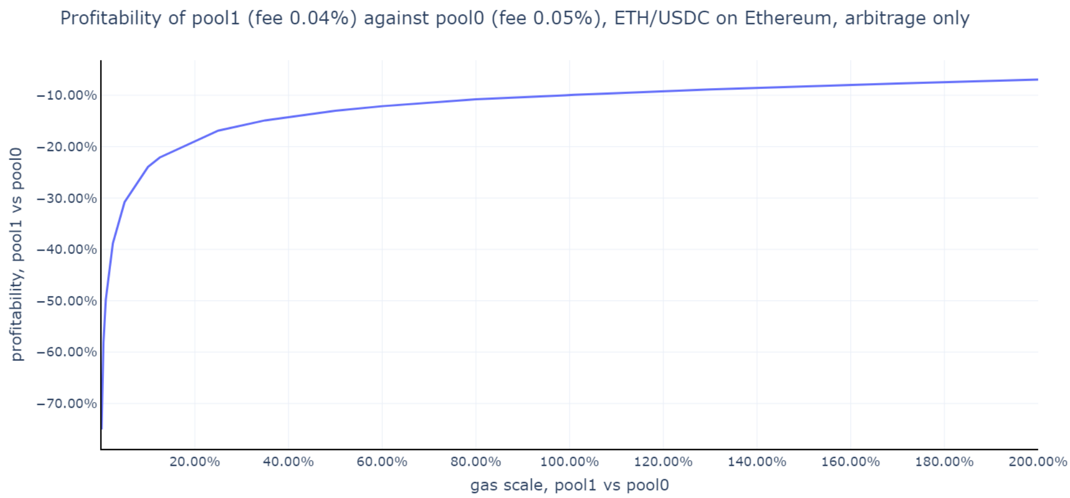


Figure 10: The result of liquidity variation for 0.04% fee ETH-USDC pool in Ethereum, arbitrage only

The behavior of the system looks different if the fee value in pool1 begins to differ from pool0. Fig. 10 shows result if there is exclusively arbitrage.

A decrease of fee value leads to a decrease in the pool's profit from arbitrage, which is further aggravated by a decrease in liquidity: arbitrageurs prefer to conduct arbitrage less often.

The situation is different when there is not only arbitrage, but also a rational uninformed flow (fig. 11).

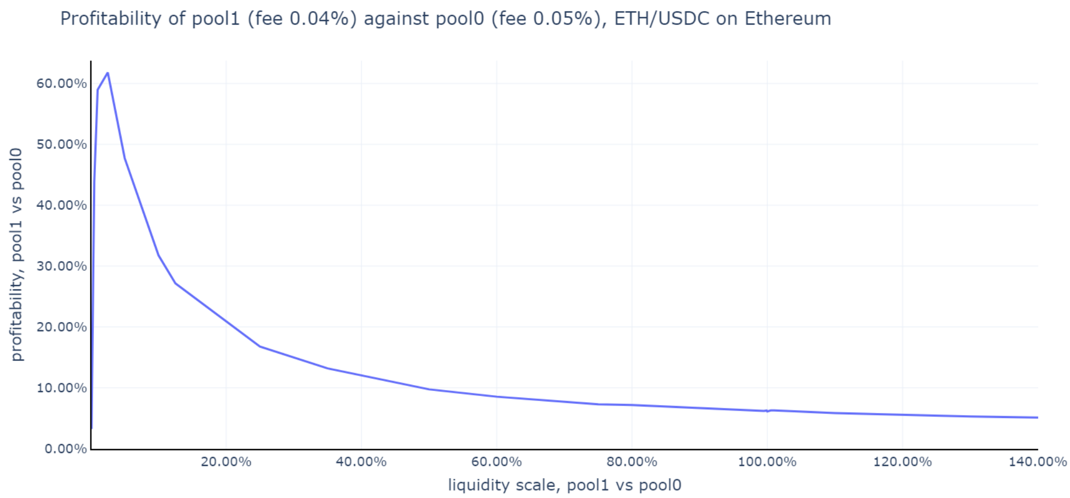


Figure 11: The result of liquidity variation for 0.04% fee ETH-USDC pool in Ethereum

This result may seem counterintuitive at first glance: decreased liquidity leads to increased profitability for providers. Nevertheless, the explanation for the observed effect is quite simple. Taking into account the results of the previous subsection (fee variation), it was determined that under the given modeling conditions, a commission of about 0.04% leads to "interception" of almost the entire random volume. Since the entire simulated market system is highly liquid, this fee change plays a much larger role than a small difference in price impact due to liquidity. For this reason, even with low liquidity, pool1 receives a significant portion of random trading, the profit from which is distributed among the less liquid providers, ultimately increasing the profit for each of them.

Thus, under certain market conditions (in fact, for this to happen, the market system must have excess liquidity and a significant volume of random trading), a smaller liquidity pool, due to lower fees, can significantly outperform a more liquid pool in terms of profitability. Next it is logical to check the result if the required conditions are not met. For this purpose, a simulation was carried out, in which the total liquidity of the system was reduced by 50 times (fig. 12).

After a strong decrease in the total liquidity of the system, price impact during a swap becomes a very significant factor in the distribution of random trading volumes. At the same time, reducing the fee when the pool's liquidity is low is no longer a recipe for success: a more liquid pool still remains more profitable.

The conducted study of the influence of liquidity volume confirms the significant importance of this parameter. In the absence of market oversaturation, increased liquidity leads to increased profitability of the protocol. On the other hand, there may be cases in which a further increase in liquidity does not make sense. Thus, when planning a protocol user acquisition strategy to maximize efficiency, it is necessary to take into account current

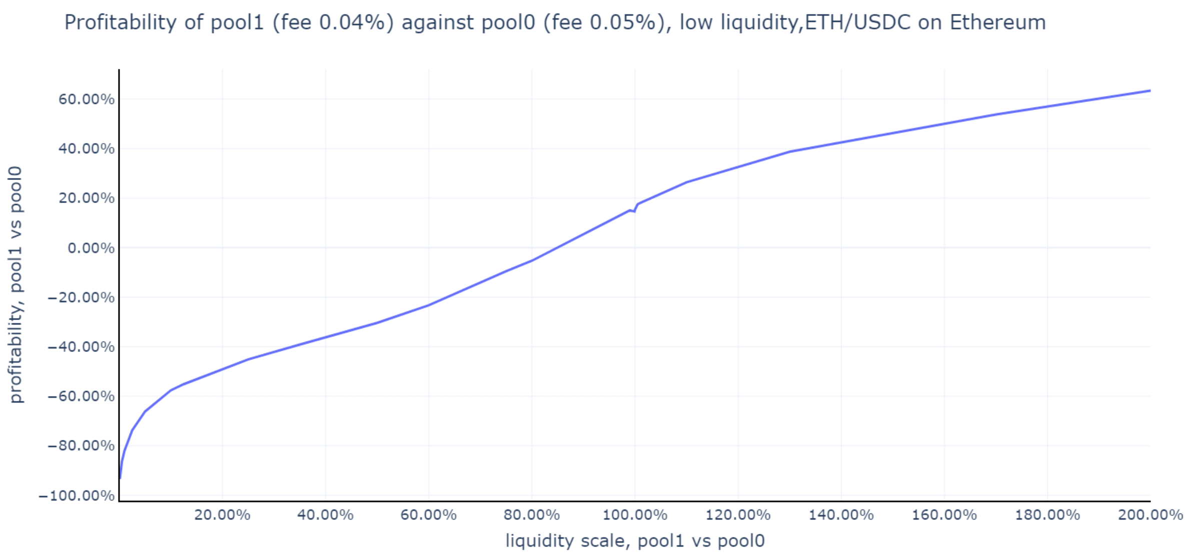


Figure 12: The result of liquidity variation for 0.04% fee ETH-USDC pool in Ethereum with low liquidity

market conditions and competitors on a given blockchain. Namely, depending on the current distribution of liquidity between protocols, adjust the commission to maximize income.

4.4 Variation of the swap cost (gas)

In this experiment, each specific simulation is performed at a given swap cost (gas per swap) in pool1 relative to pool0. The ratio is specified using the G_{scale} coefficient.

When simulating a highly liquid trading pair ETH/USDC on the Ethereum blockchain, the following result was obtained (fig. 13):

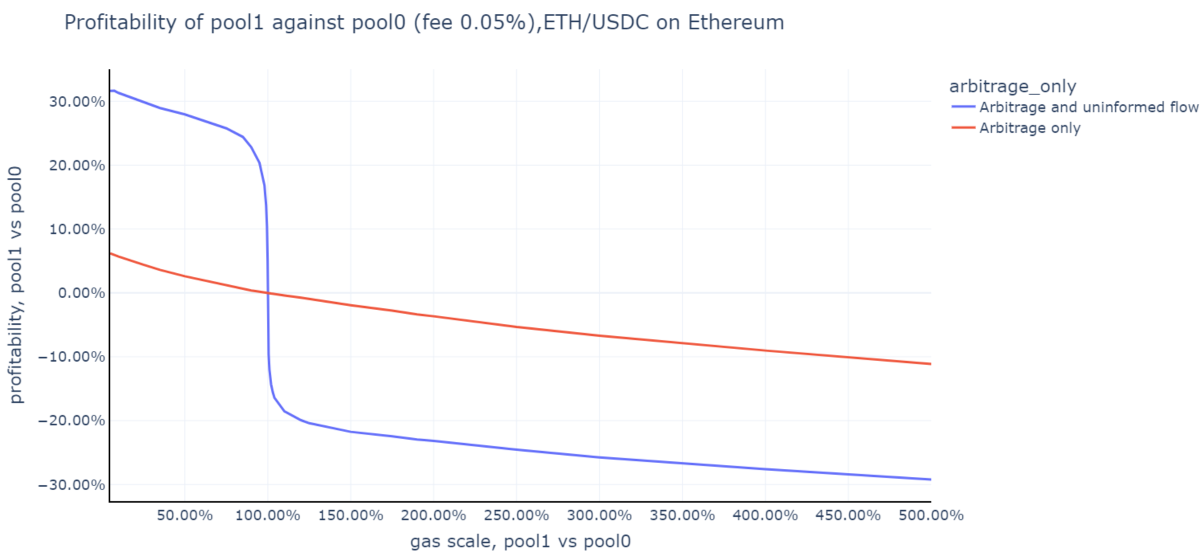


Figure 13: The result of gas consumption variation for ETH-USDC pair in Ethereum

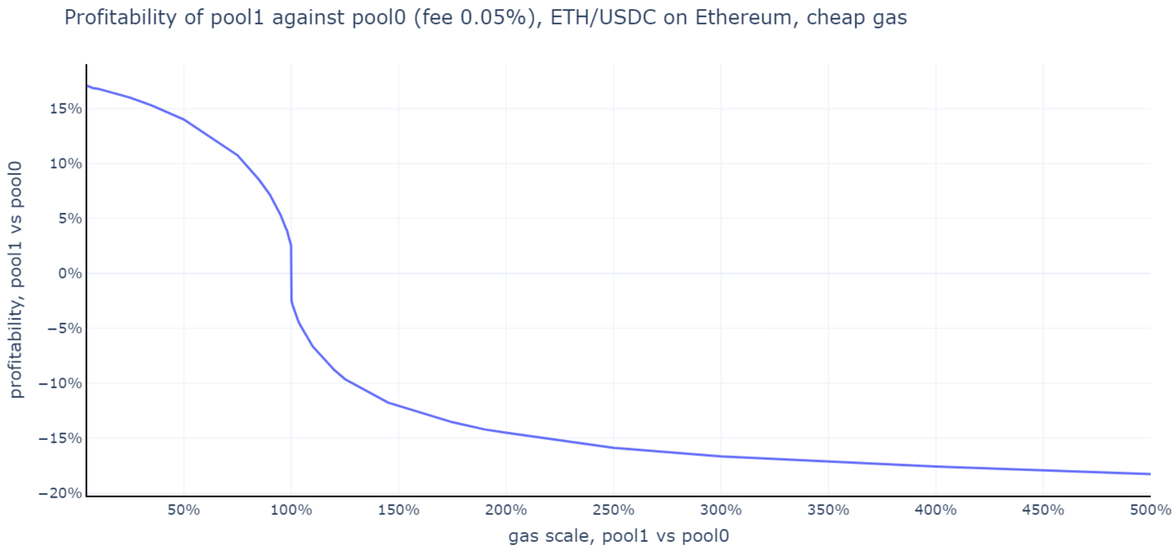


Figure 14: The result of gas consumption variation for ETH-USDC pair with cheap gas

The obtained result indicates that with expensive gas, the cost of swapping for AMM becomes an extremely important factor for a perfectly informed and absolutely rational uninformed flow. This is consistent with the trading statistics observed on real pools: for a significant share of swaps in highly liquid pairs, even a small change in gas consumption would significantly exceed the losses from price impact.

In the case of less expensive blockchains, it seems logical to assume that the cost of gas swaps will have a less dramatic impact. Indeed, with a swap cost of \$0.20, the following simulation result was obtained (fig. 14).

Although its impact is weaker with significantly cheaper gas, it nevertheless remains a significant factor influencing protocol profitability in conditions of high liquidity.

In the case of lower total liquidity (total liquidity is reduced by 50 times), the following result is observed (fig. 15):

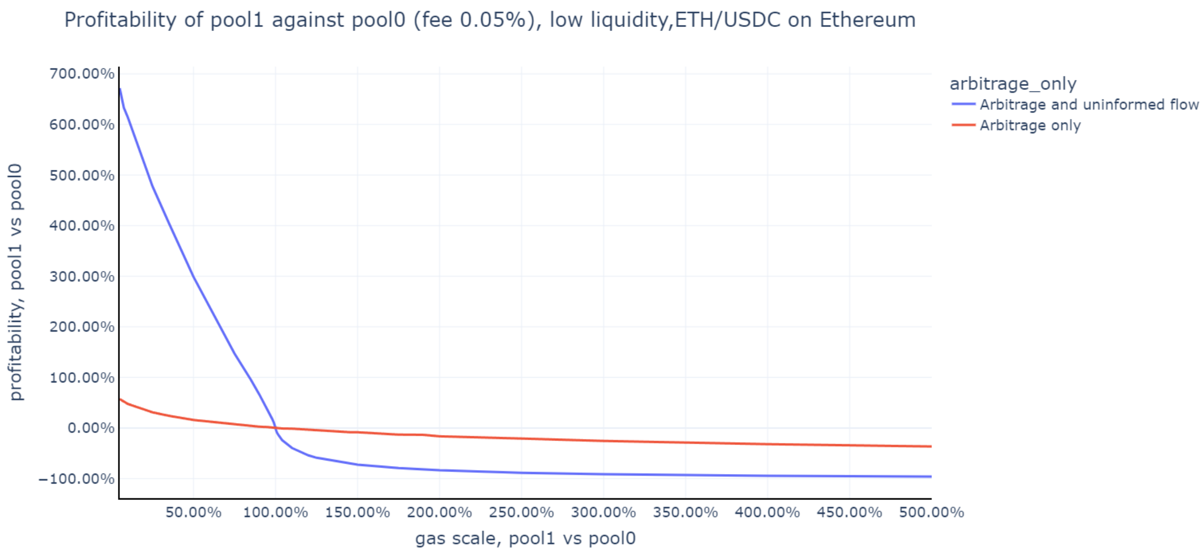


Figure 15: The result of gas consumption variation for ETH-USDC pair with low liquidity

With significantly lower liquidity, the same rational random trading volume generates greater profits for each active liquidity provider. For this reason, the significant difference in gas, while remaining an important factor, has a more dramatic impact on the profitability of AMM. On the other hand, the average income of arbitrageurs decreases when liquidity is low, which increases the importance of gas for them and forces them to use the more expensive AMM less often.

The experiment confirms the assumption that the gas efficiency of AMM is a critically important characteristic. AMM, which is more expensive to use, generates reduced returns for liquidity providers compared to competitors.

4.5 Results of a variational study of the AMM model

The experiments conducted using the simulation model are consistent with expectations and demonstrate direct dependence of the AMM profitability on the parameters under consideration. Profitability increases, on the one hand, if AMM is able to provide more favorable conditions for carrying out swaps for retail users, and on the other hand, if AMM charges a large commission on the informed trading flow (arbitrage).

At the same time, increasing the attractiveness for rational uninformed volume is achieved due to, firstly, the fastest and most accurate provision of the optimal price, and secondly, the reduction of various kinds of “side costs”. Such costs may include:

- losses due to price impact (slippage)
- gas cost per swap
- swap fee in AMM

Accordingly, by increasing liquidity, optimizing smart contracts and reducing fees, it is possible to achieve competitive advantages, providing on average more favorable conditions to end users.

In the case of externally sourced arbitrage, on the other hand, it is necessary to take into account the ratio of the expected uninformed trading volume to the informed one: increasing the fee value may increase the income of the liquidity pool from arbitrage in the presence of sufficient volatility, but at the same time radically reduce the attractiveness for uninformed order flow.

Thus, the need to take into account external factors, such as the characteristics of the assets traded and the properties of the main competing pools, together with internal factors, including the strengths and weaknesses of AMM, gives rise to the possibility of using various strategies to improve liquidity efficiency.

5 Research of approaches to fee value management

5.1 Description of the research

In the previous section, a study was conducted of the dependence of liquidity yield in AMM on some external and internal factors. The findings suggest a number of strategies that could be used to improve the effectiveness of the protocol. In addition to “simple” options, such as setting a fixed commission, attracting liquidity and optimizing smart contracts, there are also more complex strategies that require modification or addition of AMM functionality.

In this section, using a simulation model, we study the features of approaches based on automatic / manual control of the value of the swap fee. In addition, an important area of interest is the dependence of their effectiveness on market conditions.

The results of the previous experiments highlight the importance of both uninformed and informed order flow, but formulate different conditions that maximize returns from each of these sources of trading volume. For this reason, the idea arises of somehow “dividing” these flows and using informed volume discrimination to increase the protocol’s income, while maintaining its attractiveness for uninformed trading. Next, we discuss various algorithms for dynamic fees aimed at implementing the idea of separation. The final results and conclusions about their effectiveness are described at the end of this section.

5.1.1 Dynamic fee based on historical volatility

The influence of external market conditions on the effectiveness of AMM leads to the idea of reacting to them during the operation of the protocol. Depending on the types of tokens, new AMM design options were created, changing the internal invariant to provide a price curve of the required shape. On the other hand, for a standard generalized AMM, it seems reasonable to control the value of the commission based on certain characteristics of the tokens. One such characteristic is observed historical price volatility.

Based on average volatility, tokens can be divided into three classes: low volatility, medium volatility, and high volatility. Within the framework of the proposed mechanism, AMM can use specific approach for each class of tokens. Thus, low-volatility tokens carry fewer risks for liquidity providers on the one hand, and generate less informed volume on the other hand. Highly volatile pairs carry increased risk of loss for LPs, but generate more informed trading volume. Based on the mentioned features, we can suggest using a reduced fee for low volatility and a high fee for high price volatility.

It is on this approach that the dynamic commission algorithm used in the Algebra protocol of the first version is based [Algebra Protocol, 2021]. The variance of the logarithm of the pool price over the last 24 hours is used as the volatility value, with a sampling frequency corresponding to one observation every second:

$$volatility(t_{now}) = \frac{1}{window} \sum_{t=t_{now}-window}^{t_{now}} (tick(t) - tick_{average}(t))^2$$

$$window = 24 * 60 * 60$$

$$tick(p) = log_{1.0001}(price(t))$$

The resulting volatility value is used to determine which class (low, medium, high volatility) the corresponding pair of tokens belongs to. To do this, the final fee value at time t is calculated using a function of the following form:

$$fee(t) = fee_{base} + sigmoid_0(volatility(t)) + sigmoid_1(volatility(t))$$

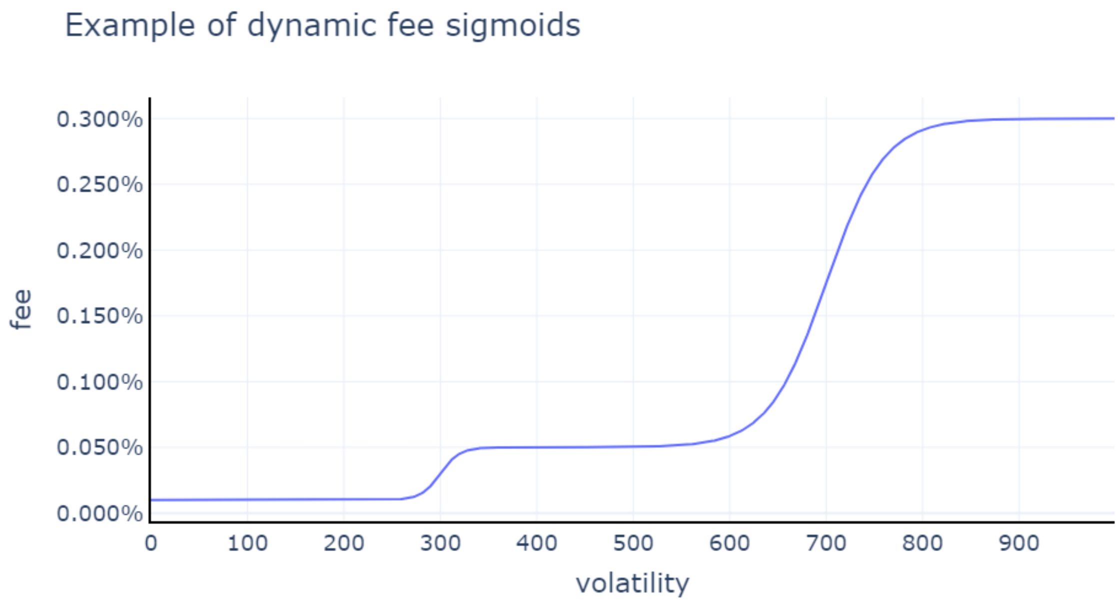


Figure 16: The example of sigmoidal fee function

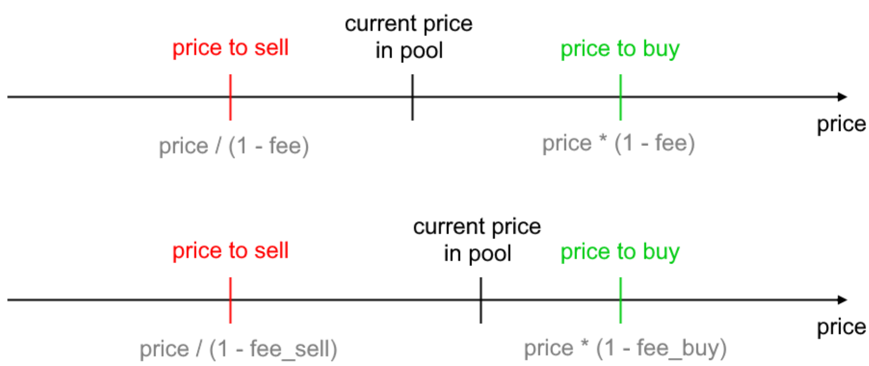
$$sigmoid_i(v) = \frac{\alpha_i}{1 + e^{\frac{\beta_i - v}{\gamma_i}}}$$

The resulting form of the volatility-based fee function is illustrated in the fig. 16.

By selecting the appropriate sigmoid parameters, the fee function can be configured in such a way that the resulting "levels" of its values correspond to the definition of low, medium and high volatility for the current token pair. As a result, the fee value will increase if trading, and with it the price movement, has been active over the last 24 hours. And go down if the price movement activity becomes less than usual.

5.2 Shifting the fee spread in the direction of price movement

Another option to take into account market conditions is to use a fee with a moving spread: the total price spread for the purchase/sale of tokens remains fixed, but shifts in one direction or another. As a result, the fee for selling begins to differ from the fee for buying.



From the point of view of an "usual" AMM user at the time of swap, such a liquidity pool is practically no different from a regular one: the "shift" of the commission spread with a constant internal price does not differ from the price shift with a constant spread (within certain limits). However, this approach allows AMM to create different conditions for buying and selling without changing the internal state of the liquidity pool (which simplifies its design).

A. Nezlobin on Twitter proposed [Alexander Nezlobin, 2023] an option for implementing this approach, based on price changes as a result of swaps. At the beginning of each block, if swaps occurred in the previous block, the commission "shifts" towards the price movement in proportion to the price impact.

Let δ be price impact after swap and k - const.

If the price moves down (token0 is exchanged for token1):

$$fee_{01}(t) = min(fee_{01}(t-1) + \delta * k, maxFee)$$
$$fee_{10}(t) = max(fee_{10}(t-1) - \delta * k, 0)$$

If the price moves upward (token1 is exchanged for token0):

$$fee_{01}(t) = max(fee_{01}(t-1) - \delta * k, 0)$$
$$fee_{10}(t) = min(fee_{10}(t-1) + \delta * k, maxFee)$$

The resulting dynamic fee management algorithm allows the liquidity pool to actively respond to price movements and more strongly discriminate against the correlated swap flow, which often corresponds to the informed toxic flow. As a result, if the trading volume is asymmetrical, the commission also becomes asymmetrical: the swap becomes more expensive in the direction of the price movement and cheaper in the opposite direction.

It should be noted that the algorithm also reacts to price movements with a delay: the commission will increase only for the next swap. For this reason, for cheaper execution of large swaps, as well as other ways to generate income, there may be strategies to manipulate this algorithm.

As part of the current modeling and simulation, the described version of the algorithm with $k = 0.75$ is used.

5.3 Elastic fee

As discussed previously, informed trading volume, generated by arbitrage for example, aims to profit from the pool's price lag. To do this, the arbitrageur buys one of the tokens in the pool at the current pool price and sells it elsewhere at the pre-known fair price of the token.

Let's look at the arbitrage process in a zero-fee pool. Let us introduce the following notation:

p_c - current instant price in the liquidity pool

p_o - current fair market price

$\Delta x, \Delta y$ - changes in token balances in the liquidity pool after the swap

Then, assuming that arbitrage is carried out in the most efficient way, the swap in the pool will be carried out in such a way that the price in the liquidity pool after arbitrage will be equal to p_o. Changes in token balances are related to the following equations:

$$\Delta y = (\sqrt{p_o} - \sqrt{p_c}) * L$$
$$\Delta x = (\frac{1}{\sqrt{p_o}} - \frac{1}{\sqrt{p_c}}) * L$$

Thus, from the point of view of the liquidity provider, the swap was carried out at a final execution price equal to:

$$|\frac{\Delta y}{\Delta x}| = |\frac{(\sqrt{p_o} - \sqrt{p_c}) * L}{\frac{1}{(\sqrt{p_o}} - \frac{1}{\sqrt{p_c}}) * L}| = \sqrt{p_o p_c}$$

This means that the real exchange execution price is equal to the geometric mean of the initial and final prices in the liquidity pool. On the other hand, the arbitrageur makes one trade at the price $\sqrt{p_o p_c}$ and another at the price p_o. From the resulting price difference, a profit arises, which is missed by the liquidity provider and given to the arbitrageur.

In the situation described above, it would be more profitable for the liquidity provider to "sell" at the price p_o, instead of $\sqrt{p_o p_c}$. This result can be achieved by using an appropriate commission value. Let ϕ be the value of the swap commission in the liquidity pool. Then (we will further consider the case of rising prices):

$$\frac{\Delta y * (1 + \phi)}{\Delta x} = p_o$$
$$\frac{\Delta y * (1 + \phi)}{\Delta x} = \sqrt{p_o p_c} * (1 + \phi) = p_o =>$$
$$(1 + \phi) = \frac{\sqrt{p_o}}{\sqrt{p_c}} =>$$
$$\phi = \frac{\sqrt{p_o}}{\sqrt{p_c}} - 1$$

The resulting commission value leads to the fact that the hypothetical arbitrage from the price p_c to the price p_o gives all the profit to the liquidity provider, and the arbitrageur does not receive any income at all.

We call the above approach to commission management elastic fee, since the value of the commission arises in response to a corresponding price shift. It should be noted that elastic fee in its current definition coincides with the concept of slip-based fee, which was proposed by the THORChain protocol [THORChain protocol, 2019]:

$$fee_{slip} = \frac{x^2 Y}{(x + X)^2}$$

X, Y - reserves of token0 and token1 tokens of the liquidity pool. By simply transforming the elastic fee described above into a representation form using virtual pool reserves, it is easy to show that it is equivalent to a slip-based fee. This approach was used by the THORChain protocol in order to increase the profitability of the pool when there are large volume swaps. However, this option for implementing an elastic fee has a significant drawback: by splitting one large swap into several smaller ones, arbitrageur can achieve significant savings on the fee paid. Due to this, the effectiveness of such a mechanism can be reduced to zero.

In order to prevent the possibility of "bypassing" the elastic fee mechanism, we modify the fee function in such a way as to ensure its additivity under certain conditions: the commission paid for two swaps must be equal to the commission for one swap equal to the sum of these swaps. To do this, we can somehow fix the starting price p_{start} (let's call it the reference price) and calculate the commission for each part of the swap (or each individual swap) as follows:

$F(p_0, p_1, L)$ - total fee amount in tokens for the "price shift" from p_0 to p_1 with liquidity L.

So:

$$F(p_N, p_{N+1}, L) = F(p_{start}, p_{N+1}, L) - F(p_{start}, p_N, L)$$

Taking into account the reasoning about the construction of the elastic fee, the function F can be defined as follows:

$$F(p_0, p_1, L) = (\frac{\sqrt{p_1}}{\sqrt{p_0}} - 1) * \Delta y$$

Given the formula $(\sqrt{p_1} - \sqrt{p_0}) * L = \Delta y$, we get:

$$F(p_0, p_1, L) = (\frac{\sqrt{p_1}}{\sqrt{p_0}} - 1) * (\sqrt{p_1} - \sqrt{p_0}) * L = \frac{(\sqrt{p_1} - \sqrt{p_0})^2 * L}{\sqrt{p_0}}$$

Further:

$$F(p_{start}, p_N, p_{N+1}, L) = L * \frac{p_{N+1} - p_N}{\sqrt{p_{start}}} - L * 2 * (\sqrt{p_{N+1}} - \sqrt{p_N}) \tag{2}$$

The resulting form of the function F (2) is the final definition of the elastic fee. Another design issue with this approach is the choice of p_{start} . We believe that choosing a price at the beginning of a block is justified: due to this, the efficiency of splitting arbitrage within a single transaction (or block) is lost, while the effect for casual users of the protocol is minimized.

Obviously, taking all profits from arbitrage has a number of negative consequences for AMM and encourages the use of strategies to circumvent this mechanism. The following subsection describes such a strategy and suggests design additions that minimize negative effects.

5.4 Elastic fee with simplified arbitrage mechanism

Given the competitive nature of arbitrage, under normal conditions the arbitrageur is interested in maximizing his profit from each transaction. As a result, not only does the income increase, but also the size of the fee that the arbitrageur is willing to pay to the miner (or block producer) for including his transaction before competitors increases.

Thus, an arbitrageur should strive to obtain maximum transaction execution income with his operations. Profit maximization is achieved at the moment when the marginal price of the arbitrated AMM, taking into account the commission, becomes equal to the external price (further price movement leads to negative profit). For a pool with an elastic commission, obtaining the appropriate price value can be done using the formulas below.

Let's consider the case when the price of AMM needs to be moved "up". In this case, the arbitrageur purchases token0 (x) using the following formulas:

$$\Delta y * (1 - fee) = (\sqrt{p_o} - \sqrt{p_c}) * L$$

$$\Delta x = (\frac{1}{\sqrt{p_o}} - \frac{1}{\sqrt{p_c}}) * L$$

$$fee = (\frac{\sqrt{p_1}}{\sqrt{p_0}} - 1)$$

Respectively:

$$\Delta y = L * (\sqrt{p_1} - \sqrt{p_0}) \frac{\sqrt{p_0}}{2 * \sqrt{p_0} - \sqrt{p_1}}$$

From here we further obtain the equation for the price of AMM after the swap:

$$\sqrt{p_1} = \sqrt{p_0} + \frac{y * \sqrt{p_0}}{L * \sqrt{p_0} + y}$$

Then the value of the commission for this swap comes down to:

$$fee = (\frac{\sqrt{p_1}}{\sqrt{p_0}} - 1) = \frac{y}{L * \sqrt{p_0} + y}$$

And the total amount of the fee paid:

$$fee_{sum} = y * fee = \frac{y^2}{L * \sqrt{p_0} + y}$$

The marginal fee value (instant fee for an infinitesimal swap) after the swap is thus obtained from the total fee in the following form:

$$fee_{marginal} = \frac{\partial fee_{sum}}{\partial y} = y * \frac{2 * L * \sqrt{p_0} + y}{(L * \sqrt{p_0} + y)^2}$$

By substituting instead of y the corresponding formula from p_0 and p_1, we can reduce liquidity and obtain an expression of the form:

$$fee_{marginal} = \frac{(a + \sqrt{p_0})^2 - p_0}{(a + \sqrt{p_0})^2} = 1 - \frac{p_0}{(a + \sqrt{p_0})^2}$$

$$a = (\sqrt{p_1} - \sqrt{p_0}) * \frac{\sqrt{p_0}}{2 * \sqrt{p_0} - \sqrt{p_1}}$$

Taking into account that the marginal price must be equal to the external price, we obtain the following equation:

$$p_{out} = \frac{p_{opt}}{1 - fee_{marginal}(p_0, p_{out})}$$

From here we ultimately get the value of p_opt:

$$p_{opt} = \frac{4 * p_0 * p_{out}}{(\sqrt{p_0} + \sqrt{p_{out}})^2} \tag{3}$$

The resulting AMM price maximizes the income from arbitrage of the pool with an elastic fee against the external price p_{out}. Given the nature of the designed elastic fee, the most profitable arbitrage strategy is to shift the AMM price in each block to the corresponding p_{opt} (3), gradually approaching p_{out}. With free transactions, this approach allows arbitrageur to get, at a limit, a certain share of the potential maximum arbitrage profit achievable in a pool without an elastic fee.

Best possible scenario for arbitrageur (without any fee) looks like:

$$p_{max} = L * (\frac{1}{\sqrt{p_0}} - \frac{1}{\sqrt{p_{out}}}) * p_{out} - L * (\sqrt{p_{out}} - \sqrt{p_0})$$

Then the ratio of the marginal profitability from pool arbitrage with an elastic fee to the maximum theoretical profitability has the following form, depending on the deviation of the AMM price from the ideal market price (fig. 17):

Figure 17: The maximal asymptotic arbitrage profit for different price deltas

From the results obtained, we can conclude that with small price deviations, an ideally rational arbitrageur is able to obtain an income close to 66.66% of the maximum (2/3). Stronger deviations allow him to get more, but also require more blocks.

The described strategy of dividing arbitrage into blocks allows the arbitrageur to minimize losses from the presence of an elastic fee. In the extreme case, if the market price does not change and there is no competition at all, the arbitrageur can almost completely neutralize the effect of the elastic commission, bringing the price closer in extremely small steps. In practice, due to the presence of competition and market price movements, the most profitable strategy is to arbitrage as quickly as possible, otherwise there is a risk of losing profits.

The resulting design of the elastic fee leads to lagging price movement. AMM lags behind the current market price. This fact may reduce the attractiveness of AMM for random uninformed trading volume, interested, on average, in the closest to the market price.

To minimize the negative effect of price lag, we propose to supplement it with a simplified arbitrage mechanism - shortcut arbitrage. When using this mechanism, the elastic fee is disabled, but the arbitrageur is obliged to transfer a fixed percentage of the income resulting from the arbitrage to the pool. As a result of using this approach, the value of the internal price in the liquidity pool reaches the market value as quickly as possible, and liquidity providers receive additional income from the informed trading flow.

From the point of view of an arbitrageur, the mechanism described above allows one to get more profit per block and reduce the risk of losing this income in competition with other arbitrageurs. The value of the coefficient of the portion of income that must be transferred to the liquidity pool after arbitration must be set in such a way as to ensure the economic feasibility of using shortcut arbitrage. As part of the current simulation experiments, a coefficient of 33% is used: the arbitrageur is exempt from the elastic fee, but is required to transfer 33% of the arbitrage income to the pool, otherwise the transaction will be reverted.

Thus, the concept of an elastic fee proposed in the previous section becomes an important part of a more complex mechanism that allows stimulating the desired behavior of arbitrageurs - the elastic fee acts as a brake that prevents arbitrage from being carried out directly, without the use of special methods and strategies.

5.5 Results of a study of mechanisms for increasing the profitability of liquidity providers

Using a simulation stand, the profitability results of the previously described commission management mechanisms were obtained:

- different fixed fee values
- volatility-based adaptive fee
- elastic fee
- elastic fee with shortcut arbitrage
- moving fee spread

Each simulation corresponds to a base fee value: this is the starting fee value (in the case of volatility-based fee, this is the minimum fee value).

5.5.1 Ethereum arbitrage only

First experiment: simulation of a highly liquid USDC/ETH trading pair on the Ethereum blockchain without any uninformed flow (arbitrage only). The following result was obtained (fig. 18):

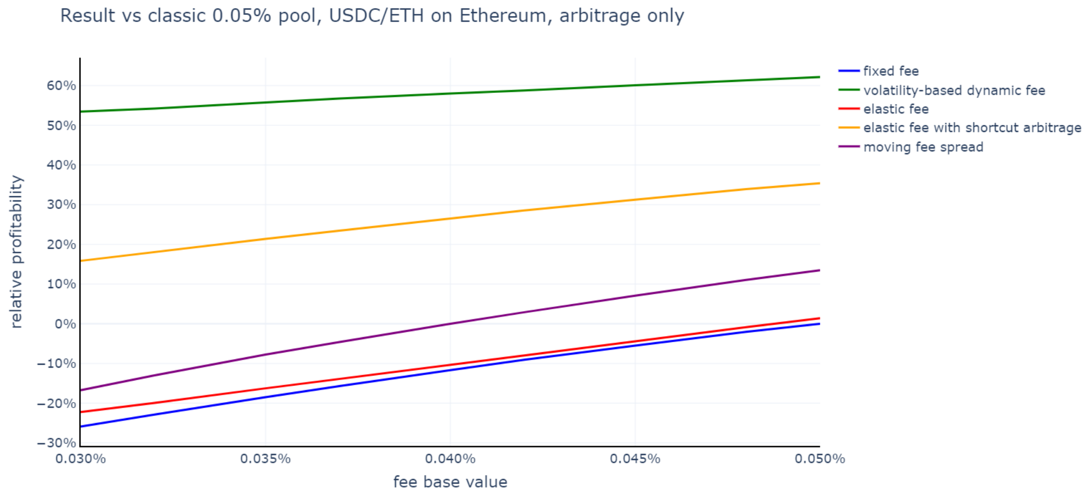


Figure 18: The result of simulation in Ethereum, only arbitrage

In this experiment, the best result is shown by volatility-based dynamic fee. This is because with the default sigmoid settings, this dynamic fee tends to set a higher fee compared to other mechanisms: most of the time the liquidity pool fee was around 0.3%. Thus, this observation is consistent with the fee variation results in the previous section: increasing the fee within certain limits increases the income of liquidity providers from arbitrage.

Elastic fee with shortcut arbitrage and moving fee spread also show good results. The elastic fee with shortcut profitability is based on the assumption that arbitrageurs are forced to give 33% of their income to the liquidity pool, which in practice may not always be the optimal strategy for them. A moving commission spread, on the other hand, causes the commission to shift in the direction of the price movement, which on average actually increases the pool's arbitrage income.

At the same time, the elastic fee in its pure form differs little from the fixed commission value: sharp and large price jumps in a given pool are not such a frequent event as to lead to a significant difference from the fixed commission.

When using a rough estimate of additional gas costs, the effectiveness of complex fee management methods slightly decreases.

5.5.2 Ethereum with uninformed flow

Completely different results are observed in the presence of not only arbitrage, but also rational uninformed flow. This experiment simulated a flow of uninformed trades with an average expected volume of $4,500 (fig. 19).

Rational uninformed flow prefers not to use pools with large fees, so volatility-based dynamic fee with standard settings demonstrates a much less outstanding result. The best profitability is observed with elastic fee with shortcut arbitrage. This option in the best configuration involves combining the benefits of discriminating against informed traders (arbitrage) with attracting uninformed ones (due to a commission of about 0.037%).

At the same time, the slightly reduced fixed fee value outperforms almost all other options in terms of profitability. It is logical to assume that the effectiveness of this approach should increase even more if we take into account the resulting excess gas costs.

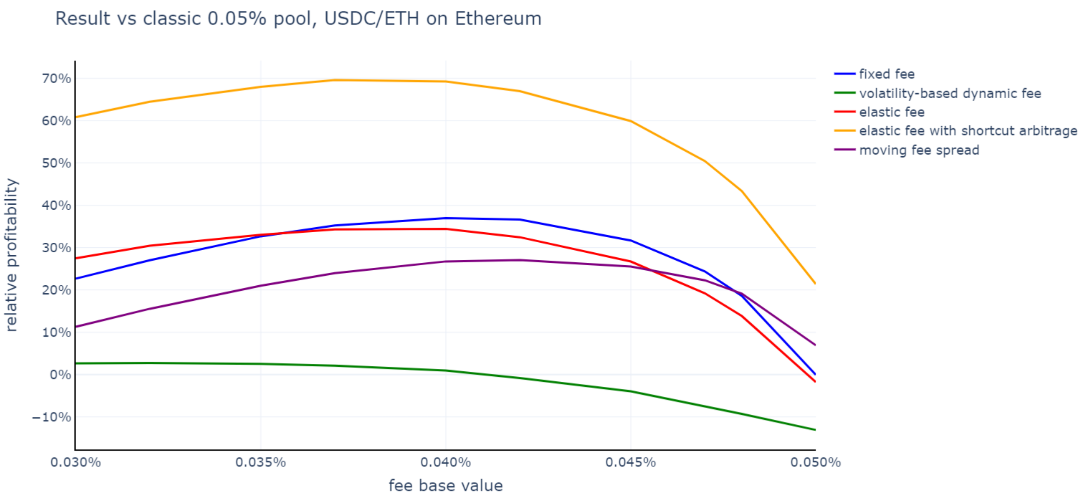


Figure 19: The result of simulation in Ethereum

This is exactly the result that is observed (fig. 20):

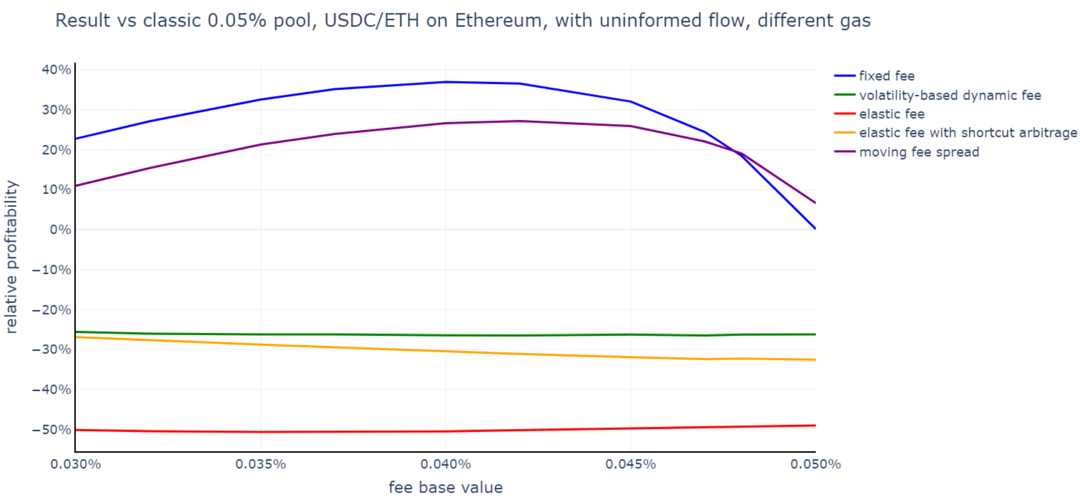


Figure 20: The result of simulation in Ethereum, with gas overhead estimations

On such an expensive network as Ethereum, increased gas consumption by the liquidity pool sharply reduces its profitability in a system with rational uninformed flow. As a result, simpler options such as changing the fixed fee or using a simple moving fee spread implementation may be much preferable to more complex automated approaches to managing fees.

Another important factor when comparing AMMs is their liquidity. If the liquidity of the simulated pool is 10 times less than the liquidity of the competitor, then the following results are observed, with the same (fig. 21) and different gas estimates (fig. 22), respectively:

To attract rational uninformed flow, a low-liquidity pool needs to reduce fees, compensating for the increased price impact. Moreover, with gas so expensive, setting a fixed low fee should be the most effective of the proposed strategies.

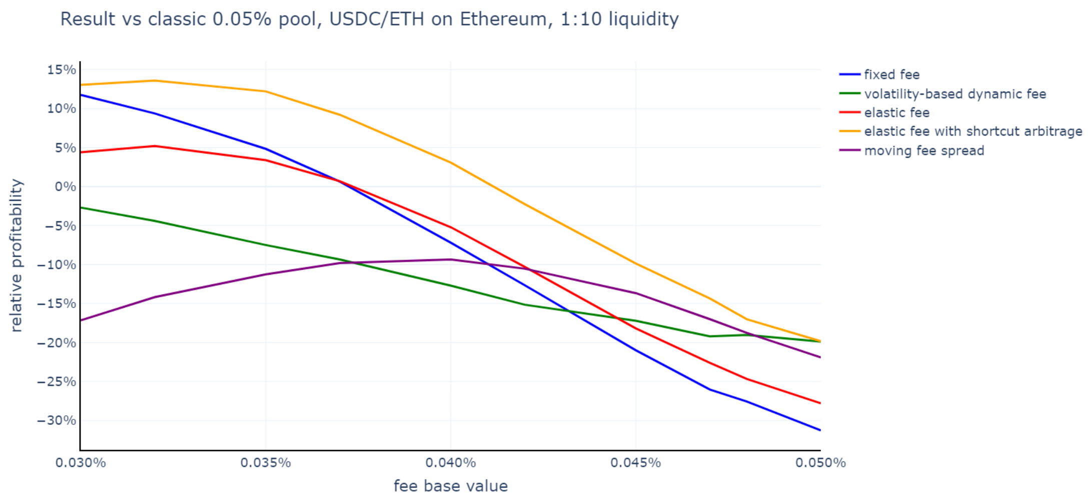


Figure 21: The result of simulation in Ethereum, with 1:10 liquidity value

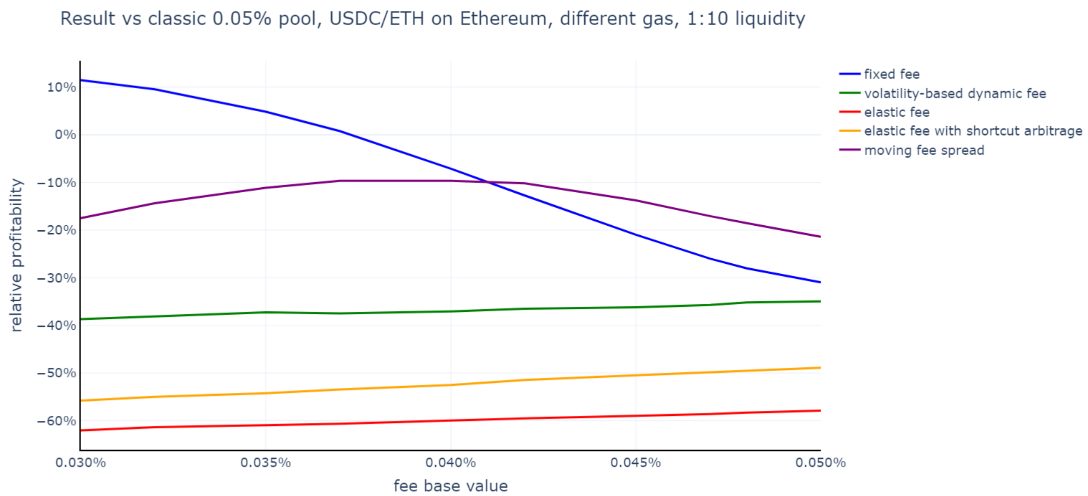


Figure 22: The result of simulation in Ethereum, with 1:10 liquidity value and gas overhead estimation

5.5.3 Polygon

The dynamics of the profitability of the pool depend, among other things, on the characteristics of the blockchain, such as the cost of gas and block time. Therefore, experiments were also carried out using the cheaper and faster Polygon blockchain as an example.

As a result of simulating the highly liquid USDC/ETH pair on the Polygon blockchain without uninformed flow, the following result was obtained (fig. 23).

Just like in the Ethereum simulation, providers' income from arbitrage increases with the fee value increase. At the same time, on average, the same fee management methods do not demonstrate such a strong gap from the fixed fee: shorter block time and cheaper gas lead to more frequent, but smaller arbitrage with a smaller price deviation.

As expected, changing gas prices has little effect on returns in a purely arbitrage simulation.

The uninformed flow simulation experiment simulated a trade flow with an expected average value of $270, which is consistent with the observed trading distribution in the

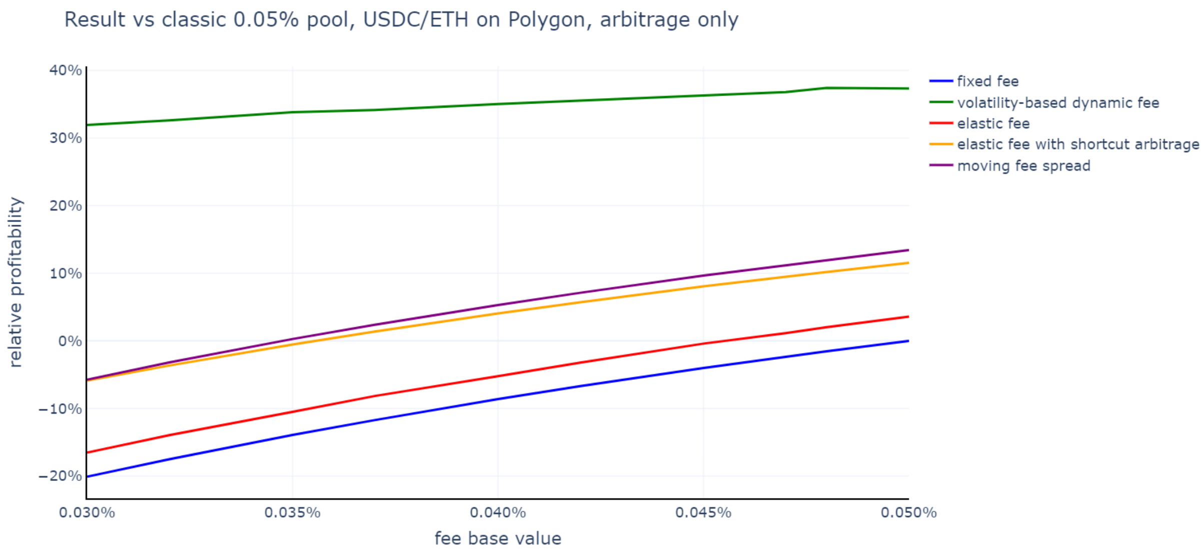


Figure 23: The result of simulation in Polygon, only arbitrage

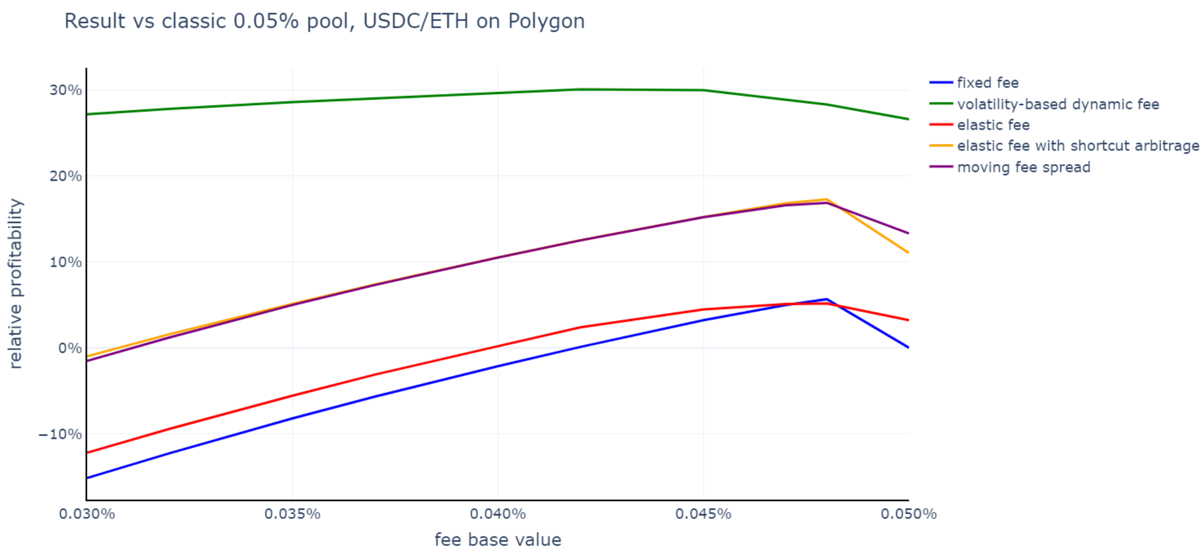


Figure 24: The result of simulation in Polygon

corresponding UniswapV3 pool on Polygon. The result differs from that obtained for Ethereum (fig. 24).

Despite the fact that volatility-based dynamic fee leads to a higher fee on average, it significantly increases the profitability of the pool. The low average value of an uninformed flow transaction leads to a significant impact of arbitrage income on the total profit of the liquidity provider. For the same reason, setting a reduced fixed commission provides a smaller increase in efficiency compared to the experiment on Ethereum.

Despite the much lower cost of gas, taking into account rough estimates of additional costs also affects the result (fig. 25):

In some cases, cheaper gas methods are also more profitable. In particular, the moving fee spread demonstrates a good result. However, more aggressive commission management using volatility-based dynamic fees shows the best results.

An experiment was also conducted with a significant difference in liquidity; the simulated

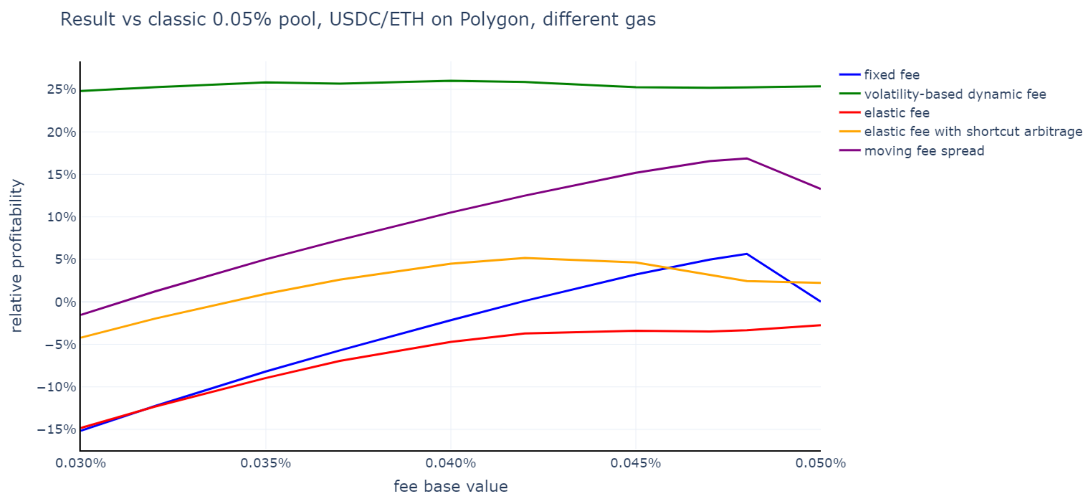


Figure 25: The result of simulation in Polygon, with gas overhead estimation

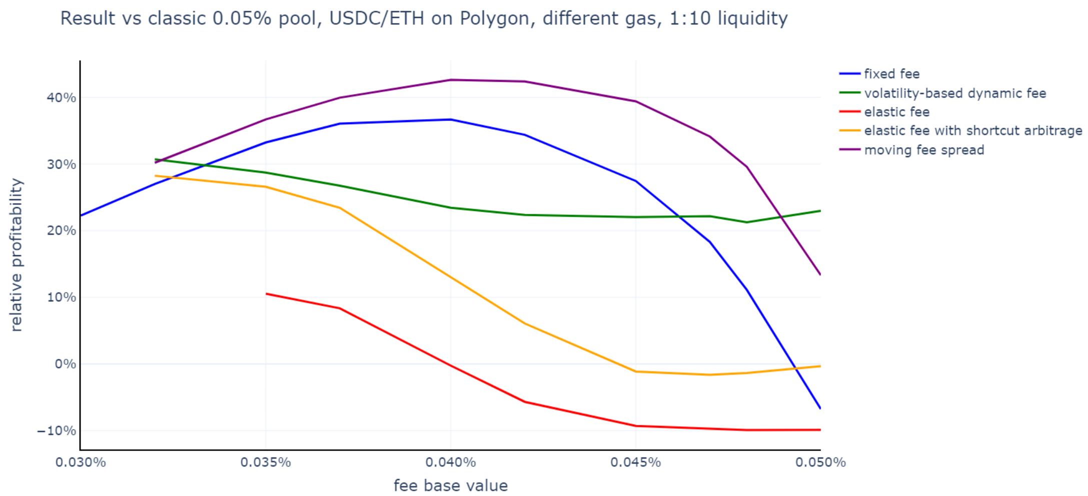


Figure 26: The result of simulation in Polygon, with gas overhead estimation and 1:10 liquidity value

pool has 10 times less liquidity than its competitor (fig. 26). The observed situation differs from the previous ones: with lower liquidity, the best results are shown by cheap approaches that preserve the attractiveness of the pool for uninformed flow.

5.5.4 Conclusions

As a result of the simulations, the following conclusions can be formulated regarding the mechanisms for managing the pool fee value:

- there is no clearly best approach for all cases
- good results, especially within expensive blockchain, are demonstrated by very simple and cheap methods, such as fixed commission and moving fee spread
- in highly liquid active pairs, elastic fee with shortcut arbitrage allows providers to get a big increase in profitability

- with the significant dominance of arbitrage, an increased value of the fixed fee may be the most profitable option

6 Discussion and applicability of the results in practice

The model used in the simulations is based on a number of assumptions. These assumptions include the absolute rationality of all trading participants. This assumption, in particular, is used to model the behavior of uninformed flow: for each swap, all available pools are compared and the best possible way to carry out the transaction is selected. In practice, such an approach is partially implemented with the help of aggregators that help users conduct transactions in the most efficient way.

However, statistics show that often the big bulk of uninformed flow enters liquidity pools not through the aggregator, but through the protocol UI. We compared the total volume of all swaps in the UniswapV3 pool on Arbitrum, at the beginning of which one or another contract (tx_to) was called. After assessing the nature of the transactions and the contracts used, the following result was obtained (fig. 27):

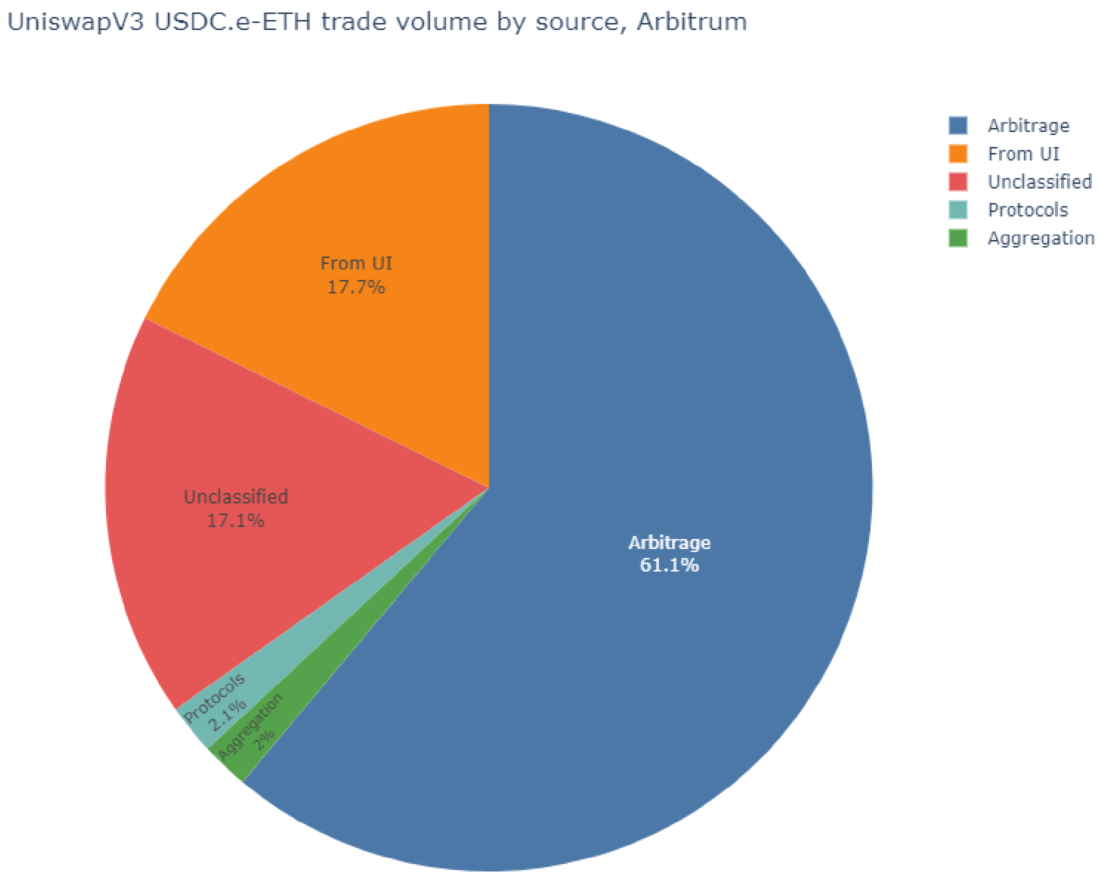


Figure 27: Estimation of volume distribution by source, USDC.e-ETH 0.05% UniswapV3, on Arbitrum

It should be noted that "Arbitrage" also includes behavior similar to MEV. The "Unclassified" category includes hundreds of different smart contracts with a small total volume. The "From UI" category includes calls directly to well-known Uniswap routers.

A similar picture is observed in the ARB-ETH pool (fig. 28):

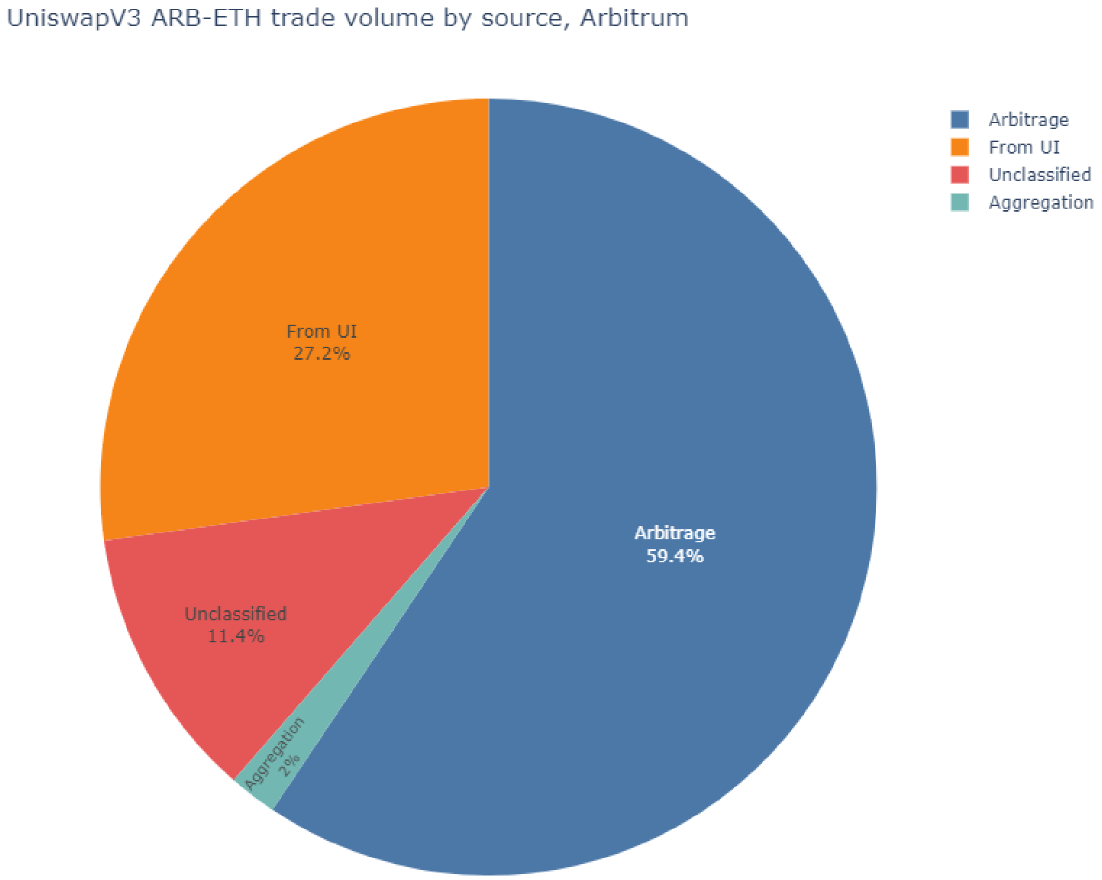


Figure 28: Estimation of volume distribution by source, ARB-ETH 0.05% UniswapV3, on Arbitrum

From the results obtained, it follows that the nature of the behavior of uninformed flow in practice differs significantly from the model we use: it is not rational and is not inclined to use aggregators to determine the most profitable way to conduct a transaction. On the contrary, a large number of retail users simply directly use one or another protocol that they are accustomed to or have the most confidence in.

Since a significant proportion of real uninformed flow consists of conditionally "loyal" volume, simulation results relying on uninformed flow are difficult to apply in practice: for example, a small reduction in commission is unlikely to lead to such a dramatic redistribution of uninformed flow as was obtained during the simulations. Accordingly, methods and approaches based on "interception" of rational uninformed flow may not be so effective.

On the other hand, the arbitrage model used is similar to the observed statistics. For this reason, methods based on informed flow discrimination can be expected to produce results similar to simulation. At the same time, one of the most profitable strategies may be to increase the commission to values that will not lead to an outflow of "loyal" volume. This assumption is confirmed, among other things, by Uniswap's successful implementation of an additional commission for using their UI: this commission exceeds (sometimes significantly) the commission in liquidity pools, but does not lead to a decrease in the trading activity of the protocol.

Other important factors for increasing the profitability of AMM remain the most efficient optimization of gas consumption by smart contracts and the attraction of greater liquidity. Reduced gas and price impact costs directly increase the income of liquidity providers.

7 Conclusions

In the course of this research, a comparison was made of the impact on AMM profitability of changes in market conditions, together with the use of various approaches to managing the value of the pool commission. The results obtained demonstrate that the following factors have a significant impact on the profitability of liquidity providers:

1. Total liquidity in the pool. Increasing overall liquidity increases profitability for all pool liquidity providers. This increase is observed until there is a "oversaturation" of the market with liquidity - an excess of demand from users and external arbitrage.

2. Gas swap cost. This factor is especially important on such expensive networks as Ethereum. Often, gas costs can exceed the impact price or even the pool commission when making a swap. Optimizing smart contracts can directly increase the profitability of liquidity providers.

3. Pool fee value. Reducing the commission relative to competitors allows pool to "intercept" more rational uninformed flow. On the other hand, an increase in commission increases the pool's income from arbitrage and "loyal" traders.

The choice of a specific blockchain has a significant impact: along with changes in transaction gas costs, the block time often also differs, the built ecosystem and the characteristics of user behavior and decentralized protocols look different. For this reason, a logical conclusion can be drawn: to select the most optimal algorithms and parameters of an AMM, it is necessary to take into account the blockchain on which this AMM will operate.

Among these factors, the fee value in the pool is the easiest to change: in addition to different fixed values, it is also possible to use more complex dynamic fee algorithms. In this article, the following commission management options are proposed and tested using simulation:

1. Fixed fee. The simplest option, which involves setting a specified commission value in the pool (including the possibility of changing it in the future). This approach is the simplest and cheapest, yet provides possibility to get good results using a lower or higher commission depending on market conditions.

2. Volatility-based dynamic fee. The Algebra protocol uses a dynamic commission based on the price volatility value over the last 24 hours. This approach is difficult to set up and incurs significant gas costs. Due to the possibility of setting a high commission, it can show good results, but the high cost of gas can offset the additional profitability.

3. Moving fee spread. A fixed fee, the spread of which "moves" relative to the price in the pool, depending on the price movement. This approach can be helpful in

discrimination against informed flow and thereby increase the profitability of the pool, without significantly reducing its attractiveness for uninformed flow. At the same time, the simplest implementation is also cheap in terms of gas.

4. Elastic fee (with shortcut arbitrage logic). A complex method of increasing the fee depending on the size of the transaction (price impact), together with the withdrawal of part of the income from arbitrage in favor of the pool. This approach is difficult to implement, but in theory it can significantly increase the income of liquidity providers by withdrawing part of the profits from arbitrage (reducing LVR).

Depending on how market conditions develop and what features a particular market and blockchain have, one or another approach, or a combination of them, may be most effective. More expensive gas forces a preference for simpler and cheaper approaches to commission management; with a lower volume of random trading in the market, methods aimed at increasing the pool's profits from arbitrage become more important; With the dominance of random trading, attracting volume with a small commission becomes more important. In this case, usually the most profitable strategies are those aimed at attracting uninformed flow along with discrimination against informed flow.

The results obtained demonstrate the strengths and weaknesses of the proposed approaches to commission management, and also formulate various options for market features that affect the profitability of AMM. The fact that maximizing profits requires an agile approach demonstrates the benefits of customizable AMM architectures such as Algebra Integral. At the moment, there is no single silver-bullet-like approach that provides guaranteed best income for liquidity providers. However, for each pool, taking into account the characteristics of the blockchain, the volume of liquidity, competitors and other market conditions, it is possible to choose the optimal solution (or solutions) that will provide the best efficiency for the pair. At Integral, we use exactly this approach: each pool is analyzed and then connected to the appropriate plugin and corresponding fee value or algorithm is used.

References

Alexander Nezlobin (2023). Retrieved 25 January, 2024 from https://twitter.com/0x94305/status/1674857993740111872.

Algebra Protocol (2021). Algebra ecosystem: Decentralized exchange. Retrieved 25 January, 2024 from https://algebra.finance/static/Algerbra%20Tech%20Paper-15411d15f8653a81d5f7f574bfe655ad.pdf.

Hayden Adams et al. (2020). Uniswap v2 core. Retrieved 25 January, 2024 from https://uniswap.org/whitepaper.pdf.

Hayden Adams et al. (2021). Uniswap v3 core. Retrieved 25 January, 2024 from https://uniswap.org/whitepaper-v3.pdf.

Jason Milionis et al. (2023). Automated market making and loss-versus-rebalancing. *arXiv preprint arXiv:2208.06046*.

THORChain protocol (2019). Slip-based fee. Retrieved 25 January, 2024 from https://docs.thorchain.org/how-it-works/fees#slip-based-fee.