Gainswap

A next-generation market maker

——Uniswap * YFI

Abstract

Gainswap is a next-generation market maker which combines the advantages of Uniswap and YFI. After adding virtual liquidity, the market maker algorithm can have a lower slippage than Uniswap. At the same time, it can use funds to manage money like YFI, and 100% of the financial income will be used to repurchase GAIN. This technical white paper explains some of the design decisions behind the Gainswap's core contracts. It also covers the main advantages of the contract, including increasing virtual liquidity and improving capital utilization,etc.

1 Introduction

Gainswap is an on-chain system of smart contracts on the Ethereum blockchain.It is decentralized, therefore it does not rely on any third parties and frees to access to any user. Gainswap also is a next-generation market maker, which utilizes the Generally Market Maker algorithm (GMM) to provide pure on-chain and contract-fillable liquidity for everyone.

Gainswap is a new implementation based on the Uniswap, with several new highlydesirable features. Gainswap has all the functions of Uniswap.First of all,it allows any user to create a trading contract for any ERC20 token. It also provides a hardened price oracle that accumulates the relative price of the two assets at the beginning of each block. This allows other contracts on Ethereum to estimate the time-weighted average price for the two assets over arbitrary intervals. Finally, it enables "flash swaps" where users can receive assets freely and use them elsewhere on the chain, only paying for (or returning) those assets at the end of the transaction. Gainswap also has all the functions of YFI. Obtain income through aggregated financial income, and use it to repurchase GAIN-Gainswap's token.

2 Features

With the success of Uniswap, AMM has proven the logical and practical correctness of liquidity on chain.However, due to the limitation of the algorithm, many problems have arisen, such as low capital utilization and large slippage.

Therefore, GMM was proposed to optimize AMM.

2.1 ERC-20 pairs

Using ETH as a mandatory bridge currency imposes costs on traders.Traders have to pay twice as much in fees as they would on a direct ABC/XYZ pair,and they suffer slippage twice.Gainswap allows liquidity providers to create pair contracts for any two ERC-20s.A proliferation of pairs between arbitrary ERC-20s could make it somewhat more difficult to find the best path to trade a particular pair,but routing can be handled at a higher layer (either off-chain or through an on-chain router or aggregator).

2.2 Prince oracle

The marginal price offered by Gainswap at time t can be computed by dividing the reserves of asset a by the reserves of asset b.

$$p_t = \frac{r_t^a}{r_t^b} \tag{1}$$

Since arbitrageurs will trade with Gainswap if this price is incorrect,the price offered by Gainswap tends to track the relative market price of the assets, This means it can be used as an approximate price oracle.

Gainswap improves this oracle functionality by measuring and recording the price before the first trade of each block (or equivalently,after the last trade of the previous block). This price is more difficult to manipulate than prices during a block.If the attacker submits a transaction that attempts to manipulate the price at the end of a block,some other arbitrageur may be able to submit another transaction to trade back immediately afterward in the same block. A miner could manipulate the price at the end of a block, but unless they mine the next block as well, they may not have a particular advantage in arbitraging the trade back.

Specifically, Gainswap accumulates this price, by keeping track of the cumulative sum of prices at the beginning of each block in which someone interacts with the contract. Each price is weighted by the amount of time that has passed since the last block in which it was updated, according to the block timestamp. This means that the accumulator value at any given time should be the sum of the spot price at each second in the history of the contract.

$$a_t = \sum_{i=1}^{t} p_i \tag{2}$$

To estimate the time-weighted average price from time t_1 to t_2, an external caller can checkpoint the accumulator's value at t_1 and then again at t_2,subtract the first value from the second, and divide by the number of seconds elapsed. (Note that the

contract itself does not store historical values for this accumulator—the caller has to call the contract at the beginning of the period to read and store this value.)

$$p_{t1},p_{t2} = \frac{\sum_{i=t1}^{t2} Pi}{t2-t1} = \frac{\sum_{i=1}^{t2} Pi - \sum_{i=1}^{t1} Pi}{t2-t1} = \frac{a_{t2}-a_{t1}}{t2-t1} \quad (3)$$

Users of the oracle can choose when to start and end this period.Choosing a longer period makes it more expensive for an attacker to manipulate the TWAP,although it results in a less up-to-date price.

One complication: should we measure the price of asset A in terms of asset B, or the price of asset B in terms of asset A? While the spot price of A in terms of B is always the reciprocal of the spot price of B in terms of A, the mean price of asset A in terms of asset B over a particular period of time is not equal to the reciprocal of the mean price of asset B in terms of asset A.For example, if the USD/ETH price is 100 in block 1 and 300 in block 2, the average USD/ETH price will be 200 USD/ETH, but the average ETH/USD price will be 150^{-1} ETH/USD. Since the contract cannot know which of the two assets users would want to use as the unit of account, Gainswap tracks both prices.

Another complication is that it is possible for someone to send assets to the pair contract—and thus change its balances and marginal price—without interacting with it, and thus without triggering an oracle update. If the contract simply checked its own balances and updated the oracle based on the current price, an attacker could manipulate the oracle by sending an asset to the contract immediately before calling it for the first time in a block. If the last trade was in a block whose timestamp was X seconds ago, the contract would incorrectly multiply the new price by X before accumulating it, even though nobody has had an opportunity to trade at that price. To prevent this, the core contract caches its reserves after each interaction, and updates the oracle using the price derived from the cached reserves rather than the current reserves.

2.2.1 Precision

The Gainswap uses a simple binary fixed point format to encode and manipulate prices. Specifically,prices at a given moment are stored as UQ112.112 numbers, meaning that 112 fractional bits of precision are specified on either side of the decimal point, with no sign. These numbers have a range of $[0, 2^{112} -1]^4$ and a precision of 2^{-112}.

The UQ112.112 format was chosen for a pragmatic reason — because these numbers can be stored in a uint224, this leaves 32 bits of a 256 bit storage slot free. It also happens that the reserves, each stored in a uint112, also leave 32 bits free in a (packed) 256 bit storage slot. These free spaces are used for the accumulation process

described above. Specifically, the reserves are stored alongside the timestamp of the most recent block with at least one trade, modded with 2^{32} so that it fits into 32 bits. Additionally, although the price at any given moment (stored as a UQ112.112 number) is guaranteed to fit in 224 bits, the accumulation of this price over an interval is not. The extra 32 bits on the end of the storage slots for the accumulated price of A/B and B/A are used to store overflow bits resulting from repeated summations of prices. This design means that the price oracle only adds an additional three SSTORE operations to the first trade in each block.

The primary downside is that 32 bits isn't quite enough to store timestamp values that will reasonably never overflow.To ensure that this system continues to function properly after this date, and every multiple of 2^{32}-1 seconds thereafter,oracles are simply required to check prices at least once per interval (approximately 136 years). This is because the core method of accumulation is actually overflow-safe, meaning that trades across overflow intervals can be appropriately accounted for given that oracles are using the proper overflow arithmetic to compute deltas.

2.3 Flash swaps

Gainswap allows a user to receive and uses an asset before paying for it,as long as they make the payment within the same atomic transaction.The swap function makes a call to an optional user-specified callback contract in between transferring out the tokens requested by the user and enforcing the invariant.Once the callback is complete, the contract checks the new balances and confirms that the invariant is satisfied. If the contract does not have sufficient funds, it reverts the entire transaction. A user can also repay the Gainswap pool using the same token, rather than completing the swap. This is effectively the same as letting anyone flash-borrow any of assets stored in a Gainswap pool (for the same 0.30% fee as Uniswap charges for trading).

2.4 Protocol fee

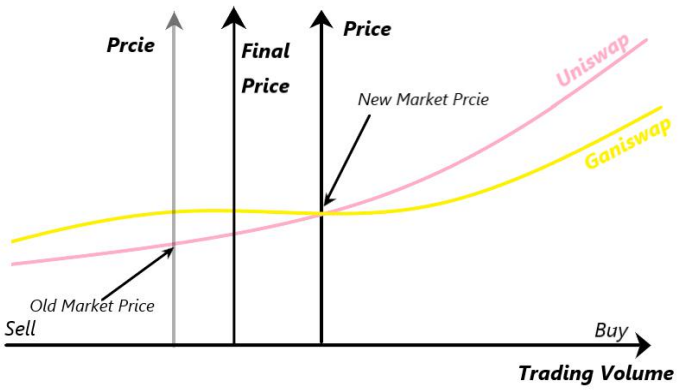
Gainswap has turned off the protocol fees to simplify transactions and put the fees in the liquidity pool to give back to the liquidity providers, which is also one of Gainswap's advantages.

3 Adavantage

After Gainswap optimizes the algorithm formula by adding the virtual liquidity. it makes the price curve more smooth in the area near the market price, and after using the funds that are not frequently traded to mine or manage wealth, the capital utilization rate is also greatly improved. The advantages of Gainswap are organized into the following aspects.

3.1 Slippage reduction

Gainswap has added virtual liquidity on the basis of the AMM algorithm formula $X * Y = K$. As the name suggests, virtual liquidity is not real liquidity, but by adding leverage to reduce the original AMM algorithm slippage loss. In addition, the introduction of leveraged X, Y, and because there are many coins in the pool, theoretically there is no virtual liquidity being used up. After adding leverage, X is nX. After adding leverage, Y is nY. The calculation formula of AMM is $(X+a) * (Y-b)=k$. The algorithm formula after adding virtual liquidity in GMM is $(nX + a) * (nY - b_1) = nnk$, where n is a value greater than 1 (usually the leverage factor). That is, on the original basis, when a value is given, since nX is greater than X and nY is greater than Y. $a/b1$ becomes smaller and the slippage becomes smaller. For example, A token is 20, B *token* is 10, $20 * 10 = 200$, that is, the k is 200. Now use 2A tokens for B tokens, according to the AMM formula (20+2) * (10-b) = 200. b is 0.91, the expected transaction price $X/Y=2$, and the actual transaction price $a/b=2.20$ Difference is 0.2. After adding 10 times leverage to add virtual liquidity, according to the GMM formula $(200+2)*(100-b_1)=20000$, at this time b_1 is 0.99. The difference between the expected transaction price $X/Y=2$ and the actual price $a/b=2.02$ is 0.02. From the calculation results that 0.02 is less than 0.2, and the slippage becomes smaller after adding virtual liquidity.



3.2 Digital currency utilization

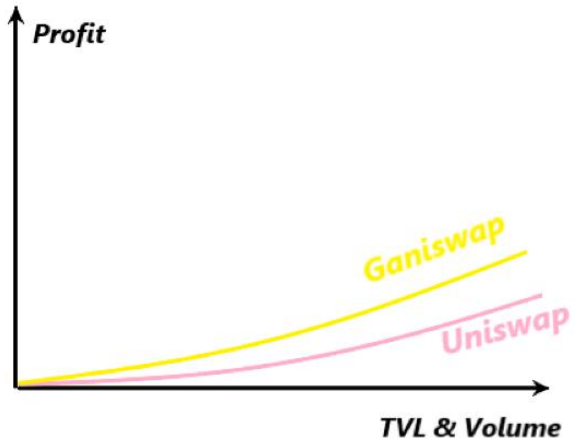
3.2.1 Flat prices in densely traded areas

As the graph shows.GMM,like AMM,provides liquidity in the price range of zero to positive infinity,but the GMM price curve is significantly flatter in the area near the oracle (market) price. Because of the use of leverage to add virtual liquidity, prices are frequently traded near the market price, rather than distributed across the entire price range.

3.2.2 Remaining funds earn income

At the same time, Gainswap proposed another method to improve Digital currency utilization that is suitable for decentralized exchanges.In the actual transaction of Uniswap, the amount of total value locked is very huge, and the actual circulation is only a small part of it, that is, most of the ETH or digital currency is only occupied but not used, so Gainswap proposes the method that Improves currency utilization. First determine the actual transaction volume in total volume locked, and then use the unused part of the total volume locked to earn stable income,such as mining.For example,first set a parameter *n*,*n* is the actual amount of liquidity occupied in the total volume locked,and the total volume locked is *TVL*, $(1-n)*TVL$ can be used for lend, exchange or earn to obtain stable income.Thereby improving the capital utilization of total value locked.

Gainswap makes full use of the trading volume(*TV*) and the total volume locked (TVL) .On Gainswap,the daily income formula is *profit = TV*0.3%+TVL*0.01%*.In Uniswap,the daily revenue fomula is *profit = TV*0.3%+TVL*0.* Therefore,on the whole,when the trading volume is the same,the profit of Gainswap is higher.

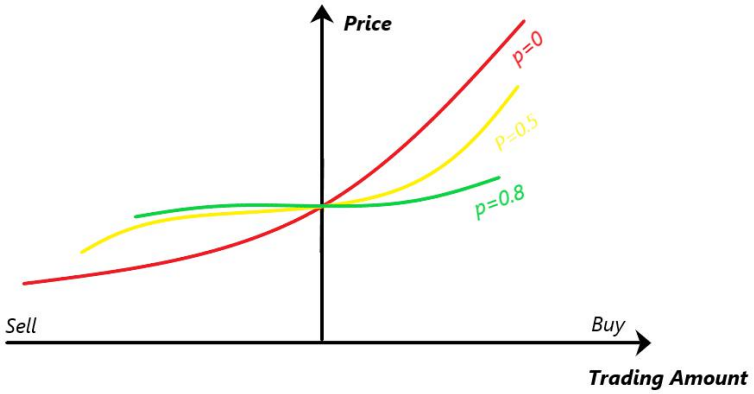


3.3 Negotiable service fee

The 30-base-point fee fixed on Uniswap can be negotiated and formulated by the community in Gainswap.That can be changed according to user needs when new situations appear.

3.4 Liquidity

The graph below is the image when the parameter ***p*** takes different values.***p*** represents the percentage of virtual liquidity to the total value locked.



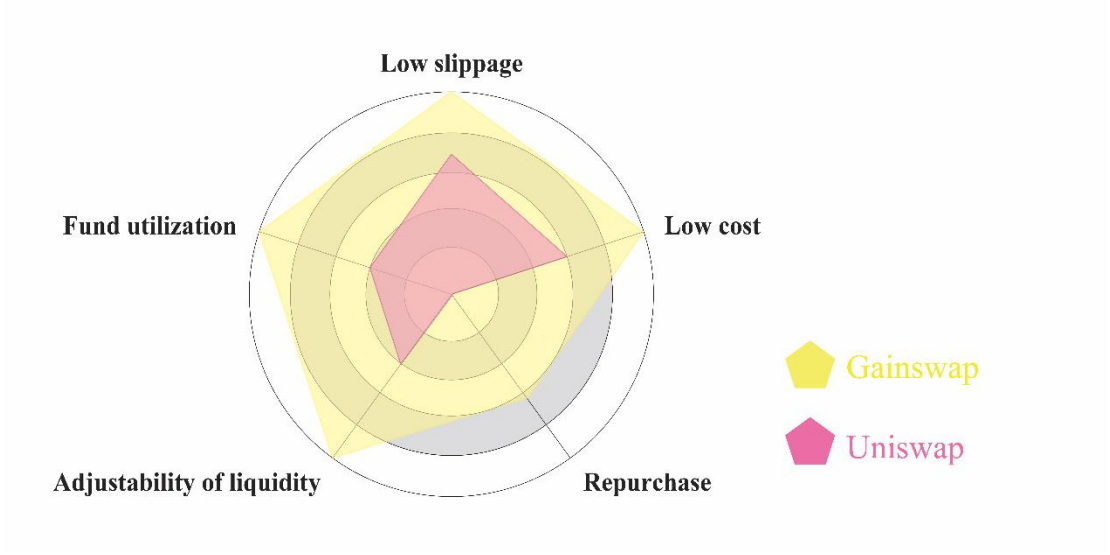
When the ***p*** value is 0, the price curve will be steep (red line), and the same as the price Uniswaps's curve. When the ***p*** becomes larger, the price curve will become flat, but the transaction price range will become smaller. This is because leverage magnifies the multiple of funds.

3.5 Repurchase agreement

Gainswap will use the proceeds from mining to repurchase GAIN. In other words, for liquidity providers who join Gainswap earlier, they will receive more rewards from Gainswap.

3.6 Comprehensive comparison

In summary, Gainswap and Uniswap are comprehensively compared from the five perspectives of cost and liquidity. The results are shown in the figure, Gainswap has a greater improvement than Uniswap in all aspects.



4 Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations. The opinions reflected herein are subject to change without being updated.

References

[1] Alan Wu: Building a Decentralized Exchange in Ethereum

[2] Fabian Vogelsteller and Vitalik Buterin. Nov.2015.URL: https://eips.ethereum.org/EIPS/eip-20.

[3] https://uniswap.io
[4] Guillermo Angeris et al. An analysis of Uniswap market 2019.arXIV: 1911.03380[q-fin.TR]