

Smart concentrated liquidity

Stefan Duprey

SwissBorg

stefan@swissborg.com

Abstract—Uniswap is the largest decentralized exchange (DEX) and one of cornerstones of Decentralized Finance (DeFi). Uniswap uses liquidity pools to provide Automated Market Making (AMM) functionality. Uniswap can provide more liquidity than its larger, centralized rivals Coinbase and Binance, because of the incentives it gives its liquidity providers to deliver better pricing to traders.

Uniswap version 3 is pioneering the new concept of concentrated liquidity feature, which allows the liquidity providers to concentrate their liquidity in a specific price range, leading to an increased capital efficiency compared to the previous v2 version.

However, the mathematical relationship between the liquidity position, the amount of assets in that position, and its price range becomes somewhat complex and the corollary to that capital efficiency is an even nastier impermanent loss risk than v2 when the initial price diverges from the initial entry price for the liquidity provision.

Research over multiple years proves that more than half of liquidity providers on uniswap v3 are losing money.

The ability to concentrate liquidity on uniswap V3 has been designed for people to gain efficiency on the capital they bring, but the key issue there is to find a suitable algorithm to rebalance the liquidity position bounds to maximize volume fees while keeping impermanent loss and rebalancing costs (transaction costs + swapping slippage) low.

Uniswap V3 liquidity provisioning requires professional expertise : the business of 'on-chain market-making' is getting more competitive and less incentives driven. Only professional quantitative market-makers will survive and thrive. Those will have risk comprehensive liquidity provisioning solution which will quickly out or rebalance the liquidity bounds according to market moves expectations.

One has to understand that the provision of liquidity in hyperbolic dexes will loose as soon as the price diverges (upwards or downwards) from where the initial price was when the liquidity was brought.

So liquidity provisioning has to be actively monitored. The chosen bounds must be actively managed to encompass the price moves and not get traversed.

In a volatile market, finding the right bounds for an optimal trading liquidity concentration is a challenging exercise : one has to find the optimal bounds rebalancing strategy for the perfect trade-off between impermanent loss, swapping costs and swapping volume fees generation.

Bullish/bearish market must be avoided at all costs : the market will traverse your bounds and leave you with your liquidity either in full stable coins in a bullish market or in full risky asset in a bearish market.

Passive liquidity investing must be seen as, in essence, a mean reverting strategy where the money is made from price fluctuation inside a specified range.

An algorithmic detection is therefore a must : one should discriminate markets as rangy, bullish or bearish and apply only passive liquidity provisioning in rangy markets.

We here detail the trade-off optimization problem to detect

liquidity bounds algorithmically.

- Maximize price inside the tightest bounds to earn volume fees (the tighter the bounds, the efficienter the capital)
- Impermanent loss and swapping slippage costs incurred at each rebalancing have to be minimized

An absolute performance analysis proves that the LP value fluctuation can be very corrosive when the underlying risky drops in value as both impermanent loss and 50% risky position asset depreciation cumulate.

The curse of Uniswap V3 liquidity provisioning is that in addition to an already nasty v3 impermanent loss, one has to add the depreciation of the 50% long side of the LP position. This leads to even greater losses in case of market crash.

We propose different algorithm flavours to mitigate the incurred losses in bearish market by :

- Long/short version : using Aave money market to short
- Market neutral version : using Aave money market to borrow the risky asset
- Hedged version : by hedging the LP position by buying an option basket for a specific maturity

We choose to present here the market neutral approach to lessen the losses due to downwards price actions and only incur market making intrinsic impermanent losses. This is the solution which is conceptually closer to real market making condition.

We use Aave money market to do so by borrowing the risky asset for the liquidity provision long side.

We here give a rebalancing methodology and the optimal matching parameters found by an exhaustive computational backtesting approach.

A proper analysis of the rebalancing costs versus fees generation is done by collecting on-chain data through 'The Graph' protocol.

I. BUILDING DIFFERENT FLAVOURS : MARKET-NEUTRAL, LONG, SHORT

Simply bringing liquidity in a Uniswap v3 position is a risky strategy. In addition to a nasty v3 impermanent loss, you can also incur the depreciation of the long side of the LP position in case of a negative price action for the risky asset.

To avoid this double penalty, one has to devise ways to take directional positions (both long and short) and take directional bets to benefit from bearish market stages too.

Or simply become market neutral by borrowing the risky asset from the money market.

A. Neutral leg

1) *Configuration*: The neutral configuration is achieved by using a money market to borrow the risky asset. We enter with a non risky/stable asset and borrow the risky part of the LP

of your borrowed Sols you use to short.

The solution is given by fixing the amount you have to short as a function of your liquidity position size ω_1 ($x = f(\omega_1)$):

$$x = \frac{(1 - \omega_1) * X_2 - \omega_1}{(1 - \omega_1) * X_2} \quad (9)$$

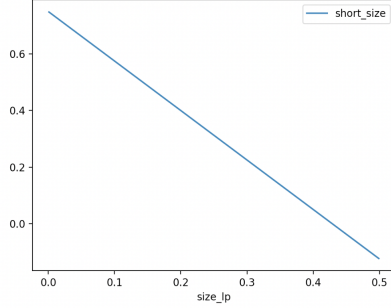


Fig. 4. Long leg

We can see that the amount of short position we want to have will directly dictate the size of our LP position and therefore the fees we produce.

For a LP position of size 0, then $x = X_2$, meaning that we short the whole ω_2 split from our asset.

For a LP of maximal size of $\omega_1 = \frac{1}{1+X_2}$ (roughly 55%), then we have no room left to short : $x = 0$.

For a LP size in between those bounds, it gives the following short position by noting : $total - short - position = S$

$$\begin{aligned} S &= x * (1 - \omega_1) X_2 * U \\ S &= ((1 - \omega_1) * X_2 - \omega_1) * U \end{aligned} \quad (10)$$

where U is the amount of the stable coins at the beginning of the epoch. The idea is to use that range of shorting size to adapt to our proprietary epoch market indicator by cumulating a long and a short leg simultaneously in an adapting ratio. By noting : $total - long - position = L$ for a long leg.

$$L = A \quad (11)$$

where A is the amount of risky coin at the beginning of the epoch.

So according to the risk level from our proprietary indicator for the next epoch, we can taylor the $\frac{short}{long}$ position ratio in a continuous way by allocating specific sizes of risky and numeraire asset:

$$\begin{cases} \frac{A}{U} = ((1 - \omega_1) * X_2 - \omega_1) & \text{market neutral} \\ \frac{A}{U} > ((1 - \omega_1) * X_2 - \omega_1) & \text{long overall} \\ \frac{A}{U} < ((1 - \omega_1) * X_2 - \omega_1) & \text{short overall} \end{cases} \quad (12)$$

II. ACTIVELY REBALANCING YOUR LIQUIDITY FOR A MARKET NEUTRAL SOLUTION

Let us here consider the market neutral configuration : we have entered an LP position with a borrowed risky asset and

are therefore only subject to the impermanent loss.

As demonstrated in the appendix D1, V3 impermanent loss can be nasty, and the tighter you encompass your price, the more violent it becomes.

We here under propose a quantitative algorithm to dynamically mitigate it.

A. Dynamical exiting

Market making is in essence a very complex topic. Professional traditional finance market makers have advanced algorithmic trading to manage their risk and avoid market trends.

Market makers profit only from rangy markets and must exit as soon as a trend takes hold (upward and downward).

V3 liquidity provisioning is market making in essence and V3 concentrated liquidity resembles an order book.

1) *Trend exiting signal*: Based on a proprietary hourly volatility/trend detecting signal, we manage to capture trendy versus rangy market hours.

This signal exhibits a very strong correlation to the hourly

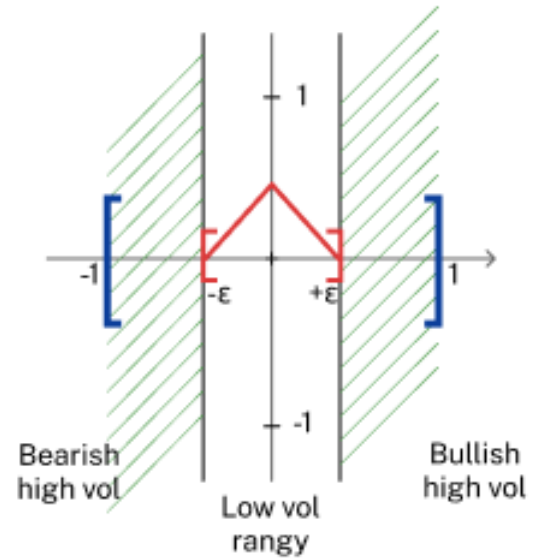


Fig. 5. On/Off Liquidity provisioning signal

price return (60% over 5 years) : $corr(S_t, R_t) > 0.5$

The idea is to use an epsilon threshold to trigger an exit signal to discriminate the hours where we will enter liquidity and the ones where we won't (methodology described in figure 5).

We here display the reconstituted performance of a strategy which would be long when we will provide liquidity provision (for a signal such that $(S_t) < \epsilon$) for different epsilons : We

Market making proportion as function of epsilon

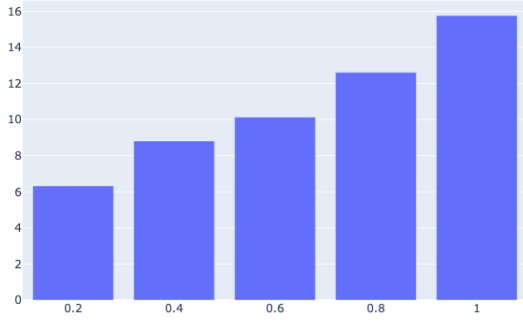


Fig. 6. Market making proportion

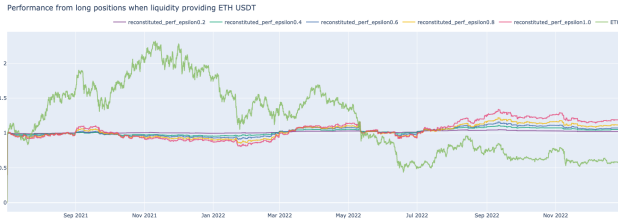


Fig. 7. Rangy market capture

see the signal is making a good job at capturing rangy market.

2) *Trend exiting signal variant*: In the sake of finding the perfect optimal trading signal, we add different variant to the plain ϵ signal :

- Early-exit triggered by $S_t < S_{t-1}$ more conservative than $S_t = 0$
- Asymmetric profiles for the LP signal trigger $0 \leq S_t < \epsilon$.

III. MODELLING GENERATED FEES

A. Dex volume historical data

We fetch the specific ETH-USDC volume on Uniswap V3 for the 0.3% fee pool using the graph protocol.

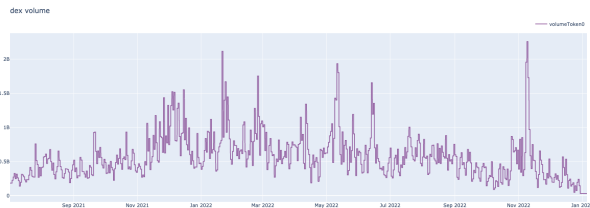


Fig. 8. Dex historical volume

B. Generated fees and v3 bounds

We take a snapshot of the APY generated by a LP position whose bounds have been placed around $x\%$ for x varying between 1% to 300%.



Fig. 9. APY as a function of bound tightness around price in %

C. Generated liquidity provisioning fees historical data

We then model the generated fees by a simple rule thumb of pro rata dex volume to get historical data.

$$APY(t) = APY(T) * \frac{DEX - VOLUME(t)}{DEX - VOLUME(T)} \quad (13)$$

This is a necessary approximation as we can't reproduce the

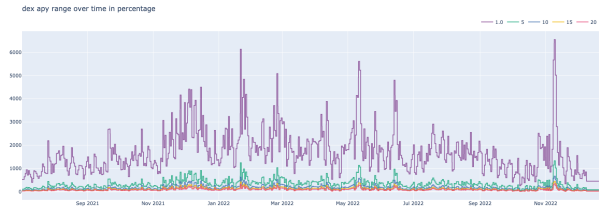


Fig. 10. Estimated over time APY as a function of bound tightness around price in %

whole past liquidity profile.

- Daily Fees = Volume * Fee * Expected Liquidity Coverage
- APY = Daily Fees Yearly Compounded

D. Encompassing the market by defining bounds

The idea behind Uniswap V3 is that the user can and must specify a price range where he wants to bring the liquidity for a better capital efficiency.

It also means that the procedure is not as straightforward as in Uniswap V2 and requires retail liquidity provider to have at least an opinion on the optimal bounds to choose.

The price range is specified by two bounds on the risky asset price $P_a \leq P_b$ and the liquidity brought by the user will only account for that range allowing a huge gain in capital efficiency.

The tighter the bounds, the more efficient the liquidity capital is, but the more you will have to rebalance and incur impermanent loss and swapping costs.

The bounds are fixed by a $x\%$ distance around the initial position entering P_0 price.

$$P_b = P_0 + x * P_0 \quad (14)$$

$$P_a = P_0 - x * P_0 \quad (15)$$

$$P_0 = \frac{P_a + P_b}{2} \quad (16)$$

Getting an idea of the proper optimal x is the topic of the next sections.

E. IL incurred at liquidity exit triggered

If we were to rebalance each time a bound is reached, we can rewrite the v3 impermanent loss according to the strategy. By noting $\kappa = \frac{P_b}{P_a} = \frac{P_0 + x * P_0}{P_0 - x * P_0}$, the v3 impermanent loss can be rewritten:

$$\frac{V_{arbitraged} - V_{held}}{V_{held}} = IL_{v2}(\tau) * \frac{1}{1 - \frac{\sqrt{\frac{2}{1+\kappa}} + \tau * \sqrt{\frac{1}{2}(1+\frac{1}{\kappa})}}{1+\tau}} \quad (17)$$

where $\tau = \frac{P_t}{P_0}$ as defined in the appendix 51. Maximum incurred impermanent loss will happen if a position is exited when one of its bound is reached. Two different maximum incurred IL loss depending on whether we reached the lower bound or the upper bound.

Upper bound reached impermanent loss:

$$\tau_{up} = \frac{P_0 + x * P_0}{P_0} > 1 \quad (18)$$

Lower bound reached impermanent loss:

$$\tau_{down} = \frac{P_0 - x * P_0}{P_0} < 1 \quad (19)$$

Equation 17 and its simplified version 56 shows how nasty v3 impermanent loss can be for tighter bounds.

But this configuration will rarely be hit, as the volatility exiting signal will most likely prevent trendy market which would lead to that case.

Most of the time, $P_t \approx P_0$ resulting in a $IL \approx 0$.

That is the whole goal of the exiting signal.

F. Market neutral costs

G. No swapping cost for the market neutral

For huge AUM (asset under management) and fully on-chain rebalancing vaults, price slippages occurring during swaps can be very detrimental to any on-chain strategy.

The market neutral version has a huge advantage : no swapping will be required.

The asset is borrowed and liquidity will only be added/removed liquidity according to our enter/exit signal.

Backtests prove that the generated fees will outperform the incurred impermanent loss (see 16 backtests).

The proportion of generated fees between ETH/USDC will depend on the statistical direction of the incurred impermanent loss.

Upwards risky asset price action will produce risky asset

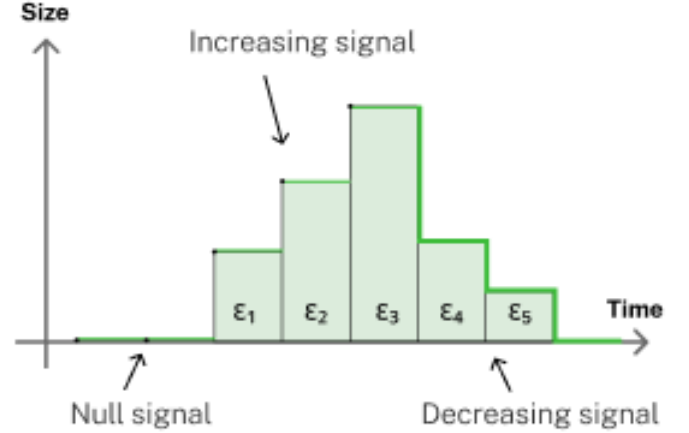


Fig. 11. LP position over time as a function of epsilon signal

generated fees.

Downwards risky asset price action will produce stable asset generated fees.

Backtests graphics 16 prove that even if 25% of the generated fees are in risky asset, will be enough to cope with impermanent loss and maintain a healthy collateral ratio on Aave.

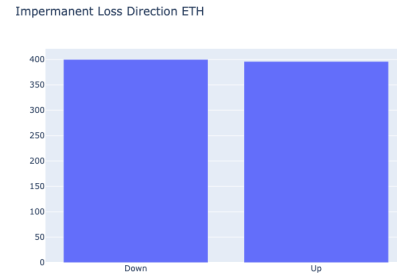


Fig. 12. Impermanent loss and price direction

We see that the repartition of impermanent losses in the backtest are as much due to price increases than price decreases.

The fee generation should therefore be balanced in between the risky asset and the stable coin.

Backtests in 16 proves then that the risky asset generated from the fees will fully compensate the assets lost from the impermanent loss.

1) *Borrowing costs*: Standard borrowing costs will be incurred on Aave money market for the market neutral solution. We here display the on-chain borrow and supply rates from

Aave on the ethereum mainnet fetched from 'The Graph' protocol.

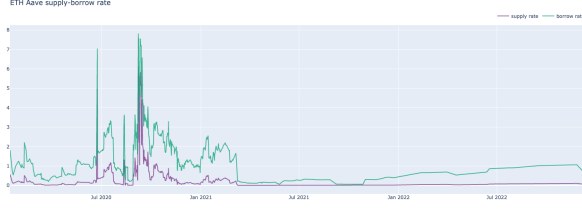


Fig. 13. Supply/Borrow costs for Ether money market neutral flavour

2) *Adding/removing liquidity costs*: The incurred cost from adding/removing liquidity is just the gas costs incurred from the two required transactions.

This fixed cost will be directly proportional to the market making time proportion as shown in 7.

We model the gas cost consumption very conservatively by pricing a transaction with the worst gas price for the last 6 month.

Chain	Arbitrum
Pool	WETH / USDC
Fees	0.05%
TVL	\$14.1m
Add Liquidity Min	0.0001
Add Liquidity Max	0.0002
Add Liquidity Avg	0.00015
Worst case Add Liquidity	\$1.95
Avg Add Liquidity at current price	\$0.19
Remove Liquidity Min	0.000064
Remove Liquidity Max	0.00015
Remove Liquidity Avg	0.000107
Worst case Remove Liquidity	\$1.46
Avg Remove Liquidity	\$0.14
Swap Min	0.000097
Swap Max	0.00015
Swap Avg	0.0001235
Worse case Swap	\$1.46
Avg swap	\$0.16

We here plot the cumulative fixed costs in dollars from the transaction costs arising from adding/removing liquidity. Those costs are fixed in the sense that they do not depend on the size of the AuM (asset under management). The bigger the AuM, the lesser those transaction costs will be.

The optimal solution of $\epsilon = 1$ gives a rough cost of a thousand dollars per year, which will become negligible as the assets under management grow.

Chain	Optimism
Pool	WETH / USDC
Fees	0.05%
TVL	\$6m
Add Liquidity Min	0.0001
Add Liquidity Max	0.0002
Add Liquidity Avg	0.00015
Worst case Add Liquidity	\$1.95
Avg Add Liquidity at current price	\$0.19
Remove Liquidity Min	0.000064
Remove Liquidity Max	0.00015
Remove Liquidity Avg	0.000107
Worst case Remove Liquidity	\$1.46
Avg Remove Liquidity	\$0.14
Swap Min	0.000097
Swap Max	0.00015
Swap Avg	0.0001235
Worse case Swap	\$1.46
Avg swap	\$0.16

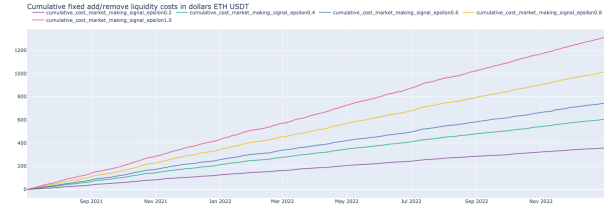


Fig. 14. Cumulative fixed add/remove liquidity transaction costs in dollars

IV. EXHAUSTIVE OPTIMIZATION

A. *Getting the optimal parameters: A trade-off between IL, transaction costs and generated LP fees*

Optimal bound distance x , volatility exiting signal ϵ must be calibrated for an optimal APY.

- Maximize price inside the tightest bounds to earn most LP fees (the tighter the bounds, the efficienter the capital)
- Calibrate liquidity provision signal epsilon : a small epsilon will bring very low impermanent loss but also very low LP fees as your market making time proportion will drastically drop.
- Add/remove transaction costs have to be minimized

We here propose an exhaustive search of the optimal parameters by running a backtest for each configuration and choosing the solution with the best Sharpe ratio.

$$\max_{(x, \epsilon)} [Sharpe(x, \epsilon)] \quad (20)$$

where $Sharpe(x, \epsilon)$ is the sharpe ratio of the back-tested path with x and ϵ as specific parameters for the algorithm.

V. RESULTS

The optimal parameters chosen after an exhaustive search are :

$$\begin{cases} \text{early} - \text{exit} = \text{True} \\ x = 3\% \\ \epsilon = 100\% \\ \text{asymmetric upward } 0 \leq S_t < \epsilon \\ \text{APY} = 358\% \end{cases} \quad (21)$$

The exhaustive search for optimal parameters gives the following optimal parameters ranked according to their annualized return.

distance	early cut	epsilon	APY	assymetric
3	True	1.0	358%	True
3	True	0.8	242%	True
3	True	0.6	166%	True
3	False	1.0	165%	True
3	True	0.4	125%	True
3	False	0.8	124%	True
3	False	0.6	96%	True
3	False	0.4	79%	True
3	True	0.2	70%	True
3	False	0.2	59%	True
5	True	1.0	46%	True
5	True	0.8	36%	True
5	True	0.6	27%	True
10	True	1.0	23%	True
5	True	0.4	20%	True

We can see from 7 that this optimal parameters will instantiate a liquidity position (or market make) for around 16% of the time.

A. Absolute performance

We here display the absolute performance graphics

We here plot the absolute performance of the strategy in

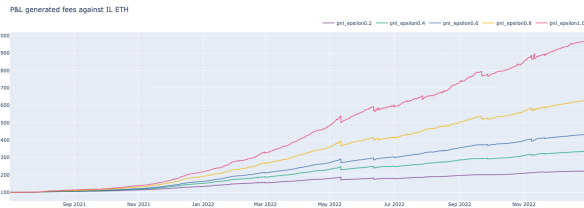


Fig. 15. Absolute performance

stable coin.

B. Generated fees

We here plot the generated fees of the strategy over time.

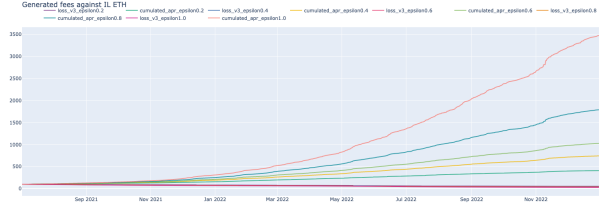


Fig. 16. Compounded generated liquidity fees versus IL v3 loss

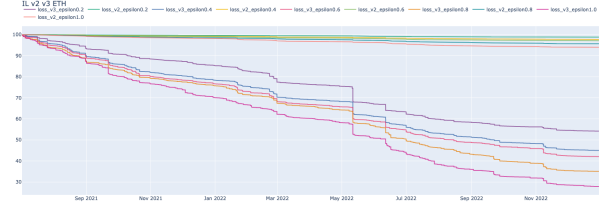


Fig. 17. Impermanent Loss v2 versus v3

C. Impermanent loss v2 versus v3

We here compare the impermanent loss v2 versus v3.

D. Drawdowns

We here plot the different ϵ drawdowns over time.

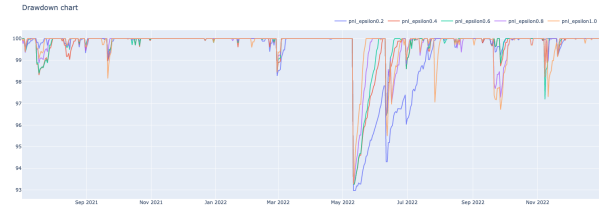


Fig. 18. Drawdowns over time

VI. HEDGING THE POSITION

Results of section V clearly expose that the biggest losses for that strategy in absolute performance are when the risky underlying price drops. If it drops enough to trigger a rebalancing, then we have a concurrence of impermanent loss, risky asset depreciation and swapping costs which can ruin multiple months of fees generation.

We therefore here propose to modulate the strategy by using proprietary trend following forecasting indicator to give a downside risk level for the epoch to come.

According to that indicator, we here under propose two different methodologies.

1) *Hedging with options instead of a short leg*: Replicating the LP payoff We here investigate another way of mitigating our downside risk when assessing downside risky epoch.

We here propose to replicate the *LP* value payoff at the end of an epoch by a basket of options whose maturity matches the end of the epoch.

We then optimize the options allocation to exactly fit the LP value payoff.

This can easily be done by an optimization algorithm minimizing the *L2* norm between the LP payoff and the derivatives payoff.

We fetch from Derebit all options for a specific maturity. For each strike, we compute long/short versions of call/put payoffs:

$$\begin{aligned} \text{Payoff Long Call} &= \max(0, P_f - K) \\ \text{Payoff Short Call} &= -\max(0, P_f - K) \\ \text{Payoff Put Call} &= \max(0, K - P_f) \\ \text{Payoff Long Call} &= -\max(0, K - P_f) \end{aligned} \quad (22)$$

For a basket of weights $\theta = (\theta_i)$, we compute the final payoff:

$$\text{Payoff}(\theta, P_f) = \sum_{\theta_i \text{ in } \theta} \theta_i * \text{payoff}_{\theta_i}(P_f) \quad (23)$$

We then minimize the functional of the *L2* distance between the two payoffs:

$$J(\theta) = \sum_{P_f \text{ in } [0, +\infty[} (\text{Payoff}(\theta, P_f) - LP_{PnL}(P_f))^2 \quad (24)$$

This loss function can be optimized via Stochastic Gradient Descent. Each weight can be positive or negative. A positive value is a long position in the options contract, while a negative value is a short position.

We here under give an example of a replicated LP PnL at a one month expiry. Analyzing the hedging cost We can then



Fig. 19. Payoff replication using options

analyse the total hedging cost by getting the options premium for Derebit and netting between the long and short premiums.

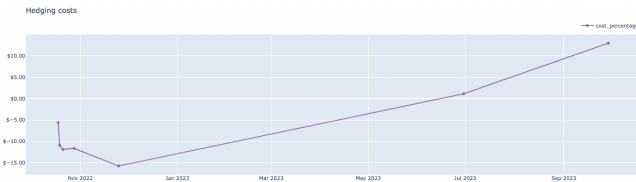


Fig. 20. Hedging costs

VII. SOLIDITY IMPLEMENTATION

A set of Solidity Smart Contracts allows us to execute the different flavoured strategies. The main feature they provide are : - managing Uniswap v3 position in atomic way - segregating the fund Manager from the Creditors

A. Managing Uniswap v3

Uniswap v3 allows for more flexible and efficient trading strategies by introducing concentrated liquidity, which allows liquidity providers to focus their liquidity within a specific price range. The protocol also allows for multiple fee tiers and customizable fees, giving liquidity providers more control over their returns. Additionally, Uniswap v3 NFTs represent a user's liquidity position in a particular pool, which provides important information about the position. In order to manage those different specificity, we built a Solidity Wrapper on top of an Arrakis V2 Vault. Arrakis V2 Vaults, allow for the creation of multiple liquidity positions in a single Uniswap v3 pool, as well as the ability to LP across multiple fee tiers under one position. That way, we can easily manage different positions at the same time, removing the need to manage multiple NFT and more importantly we can make Uniswap v3 positions fungible.

Arrakis V2 vaults is the AMM market-maker's backbone. Nonetheless, in order to fulfil our needs, we had to develop a Wrapper on top Arrakis V2 Vaults that allows to : - make atomic transactions when adding / removing transactions - enforce the rules under which the Manager has the ability to operate

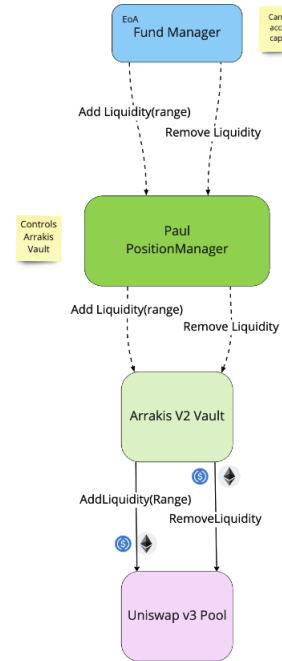


Fig. 21. Smart Contracts architecture

VIII. CONCLUSION

<https://medium.com/@RoboVault/delta-neutral-strategy-deep-dive-ae91d309b504> <https://arxiv.org/abs/2208.03318>
<https://arxiv.org/pdf/2106.14404.pdf>
<https://uniswapv3.flipsidecrypto.com/>
<https://uniswapoptimizer.com/> <https://app.flipsidecrypto.com/velocity/queries/efed04575edc-46fa-ad7b-cffd01d5b93d>
<https://app.flipsidecrypto.com/velocity/queries/bb47119b-a9ad-4c59-ac4d-be8c880786e9>
<https://arxiv.org/pdf/2106.12033.pdf>
<https://medium.com/auditless/impermanent-loss-in-uniswap-v3-6c7161d3b445>

APPENDIX

A. Hyperbolic equation

Let X denote the reserve of the risky asset (Ether, WBtc, SOL) and Y denote the reserve of the non risky stable coin (USDT, USDC, BUSD, ..). Uniswap pioneering approach was to force the reserve quantities on the pool to live on an hyperbole, hence the denomination hyperbolic dex.

$$X * Y = cte = X_{t0} * Y_{t0} \quad (25)$$

The constant is fixed by the initially brought amount at instant t_0 . The risky asset price X in stable coins Y is directly linked to the pool reserve:

$$P_{X \text{ in } Y} = \frac{Y}{X} = P_X \quad (26)$$

Basically if you have one Ether ($X = 1$) and 4500 USDT in the pool ($Y = 4500$), the ether price is $\frac{4500}{1}$. And vice versa

$$P_{Y \text{ in } X} = \frac{X}{Y} = P_Y \quad (27)$$

By denoting $L^2 = X_{t0} * Y_{t0}$ (L can be seen as the part of the liquidity brought by a single asset of the pool), a quick computation gives:

$$L = \sqrt{X * Y} \quad (28)$$

$$X = \frac{L}{\sqrt{P_X}} \quad (29)$$

$$Y = L * \sqrt{P_X} \quad (30)$$

B. Uniswap V2 Impermanent loss

Uniswap V2 liquidity pools are actively managed by arbitraging bots/traders which swap the proper quantities to readjust the pool quantities to match the consensus price formed over all exchanges (centralized and decentralized).

Let's denote t_1 an instant where the pool quantities are denoted $X_{t_1}, Y_{t_1}, P_{X,t_1}, P_{Y,t_1}$. Let's denote $t_2 > t_1$ a subsequent instant where the pool quantities are denoted $X_{t_2}, Y_{t_2}, P_{X,t_2}, P_{Y,t_2}$.

Let's imagine two different states of the reality. A first one where the pool has not been arbitrated and a second one where the pool has been arbitrated.

The quantities at instant t_2 when the pool has been arbitrated match the consensus price $P_{consensus}$:

$$P_{X_{t_2}} = \frac{Y_{t_2}}{X_{t_2}} = P_{consensus} \quad (31)$$

If the pool has not been arbitrated, its reserve quantities have not moved: they stayed at $X_{t_1}, Y_{t_1}, P_{X,t_1}, P_{Y,t_1}$.

$$P_{X_{t_1}} = \frac{Y_{t_1}}{X_{t_1}} \neq P_{consensus} \quad (32)$$

Both state values of the pool in stable coins Y can be expressed as :

$$Y_{t_1} = X_{t_1} * P_{consensus} + Y_{t_1} = X_{t_1} * P_{X_{t_2}} + Y_{t_1} \quad (33)$$

$$Y_{t_2} = X_{t_2} * P_{consensus} + Y_{t_2} = X_{t_2} * P_{X_{t_2}} + Y_{t_2} \quad (34)$$

A quick computation gives the impermanent loss formula (the risk of loss for a non arbitrated pool whose risky asset price has moved :

$$\frac{v_{t_2} - v_{t_1}}{v_{t_1}} = \frac{2 * \sqrt{\frac{P_{X_{t_2}}}{P_{X_{t_1}}}} - 1}{1 + \frac{P_{X_{t_2}}}{P_{X_{t_1}}}} = \frac{2 * \sqrt{P_{X_{t_1}} P_{X_{t_2}}} - (P_{X_{t_1}} + P_{X_{t_2}})}{P_{X_{t_1}} + P_{X_{t_2}}} \quad (35)$$

This function is symmetrical in P_{X_1} the price of the non-arbitrated pool at instant t_1 and $P_{consensus} = P_{X_2}$ the consensus price at t_2 .

The proof is straightforward :

$$\frac{v_{t_2} - v_{t_1}}{v_{t_1}} = \frac{\frac{v_{t_2} - v_{t_1}}{X_{t_1}}}{\frac{v_{t_1}}{X_{t_1}}} = \frac{\frac{v_{t_2}}{X_{t_1}} - (P_{X_{t_1}} + P_{X_{t_2}})}{P_{X_{t_1}} + P_{X_{t_2}}}$$

because $\frac{v_{t_1}}{X_{t_1}} = P_{X_{t_1}} + P_{X_{t_2}}$.

$$\frac{v_{t_2}}{X_{t_1}} = \frac{X_{t_2} * P_{X_{t_2}} + Y_{t_2}}{X_{t_1}}$$

$$\frac{v_{t_2}}{X_{t_1}} = \frac{2 * Y_{t_2}}{X_{t_1}} = \frac{2 * L * \sqrt{P_{X_{t_2}}}}{X_{t_1}}$$

$$\frac{v_{t_2}}{X_{t_1}} = \frac{2 * L * \sqrt{P_{X_{t_1}} P_{X_{t_2}}}}{P_{X_{t_1}} * X_{t_1}}$$

$$\frac{v_{t_2}}{X_{t_1}} = \frac{2 * L * \sqrt{P_{X_{t_1}} P_{X_{t_2}}}}{\sqrt{P_{X_{t_1}}} * X_{t_1}} = 2 * \sqrt{P_{X_{t_1}} P_{X_{t_2}}}$$

By renaming $\frac{P_{X_{t_2}}}{P_{X_{t_1}}} = \tau$ We can rewrite the impermanent loss equation :

$$\frac{v_{t_2} - v_{t_1}}{v_{t_1}} = \frac{2 * \sqrt{\tau}}{1 + \tau} - 1 = IL_{v2}(\tau) \quad (36)$$

- 1.25x price change results in a 0.6% loss relative to HODL
- 1.50x price change results in a 2.0% loss relative to HODL
- 1.75x price change results in a 3.8% loss relative to HODL
- 2x price change results in a 5.7% loss relative to HODL
- 3x price change results in a 13.4% loss relative to HODL
- 4x price change results in a 20.0% loss relative to HODL

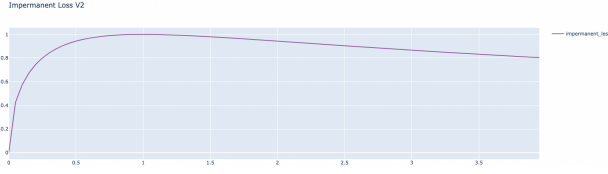


Fig. 22. Impermanent loss versus price ratio

•5x price change results in a 25.5% loss relative to HODL
“N.B. The loss is the same whichever direction the price change occurs in (doubling in price results in the same loss as halving).

C. Uniswap V3 Impermanent loss

The idea behind Uniswap V3 is that the user can specify a price range where he wants to bring the liquidity. The price range is specified by two bounds on the risky asset price $P_a \leq P_b$ and the liquidity brought by the user will only account for that range allowing a huge gain in capital efficiency.

1) *Capital efficiency and uniswap v3*: From now on, the only price we will deal with is the price of the risky asset in stable coin.

$$P = P_X = P_{X \text{ in } Y} = \frac{Y}{X} \quad (37)$$

With that notation, it comes :

$$L = \sqrt{X * Y} \quad (38)$$

$$X = \frac{L}{\sqrt{P}} \quad (39)$$

$$Y = L * \sqrt{P} \quad (40)$$

2) *Swapping inside a tick*: Inside a tick, everything works as with the previous Uniswap V2 protocol with virtual reserves.

$$X_{virtual} * Y_{virtual} = L^2 \quad (41)$$

And the reserves evolution are dictated by

$$\Delta \sqrt{P} = \frac{\Delta Y}{L} \quad (42)$$

$$\Delta \frac{1}{\sqrt{P}} = \frac{\Delta X}{L} \quad (43)$$

The smart contract will only track the L and \sqrt{P} . The reserve will be updated accordingly.

The risky asset reserve matching the highest price for which the liquidity provider is ready to provide liquidity :

$$X_b = \frac{L}{\sqrt{P_b}} \quad (44)$$

The stable coin reserve matching the lowest price for which the liquidity provider is ready to provide liquidity :

$$Y_a = L * \sqrt{P_a} \quad (45)$$

$$\begin{cases} X_{virtual} = X_{real} + \frac{L}{\sqrt{P_b}} \\ Y_{virtual} = Y_{real} + L * \sqrt{P_a} \end{cases} \quad (46)$$

The hyperbole equation thus becomes:

$$(X_{real} + \frac{L}{\sqrt{P_b}}) * (Y_{real} + L * \sqrt{P_a}) = L^2 \quad (47)$$

From now on, we will give up the real tag and call $X = X_{real}$ and $Y = Y_{real}$.

The solution is given by the three different regimes according to the current price P location regarding the liquidity bounds:

$$\begin{cases} \begin{cases} X = 0 \\ Y = L * (\sqrt{P_b} - \sqrt{P_a}) \end{cases} & , P \geq P_b \\ \begin{cases} X = L * (\frac{1}{\sqrt{P}} - \frac{1}{\sqrt{P_b}}) \\ Y = L * (\sqrt{P} - \sqrt{P_a}) \end{cases} & , P_a \leq P \leq P_b \\ \begin{cases} X = L * (\frac{1}{\sqrt{P_a}} - \frac{1}{\sqrt{P_b}}) \\ Y = 0 \end{cases} & , P \leq P_a \end{cases} \quad (48)$$

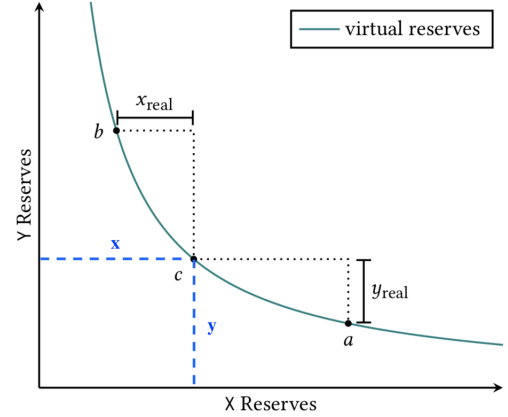


Fig. 23. Concentrated liquidity between a and b

D. Computing the impermanent loss

The value of the pool at a time t when the pool is arbitrated :

$$V = X * P + Y \quad (49)$$

$$V = L * (\frac{1}{\sqrt{P}} - \frac{1}{\sqrt{P_b}}) * P + L * (\sqrt{P} - \sqrt{P_a})$$

$$V = 2 * L * \sqrt{P} - L * (\sqrt{P_a} + \frac{P}{\sqrt{P_b}}) \quad (50)$$

Let's define $P_{consensus}$, the new price coming from a market consensus and $\tau > 0$ the price ratio.

$$P_{consensus} = \tau * P \quad (51)$$

The value of the arbitrated pool with the new consensus price is then :

$$V_{arbitrated} = 2 * L * \sqrt{\tau * P} - L * (\sqrt{P_a} + \frac{\tau * P}{\sqrt{P_b}}) \quad (52)$$

$$V_{held} = X * P_{consensus} + Y \quad (53)$$

$$V_{held} = L * \left(\frac{1}{\sqrt{P}} - \frac{1}{\sqrt{P_b}} \right) * P_{consensus} + L * (\sqrt{P} - \sqrt{P_a}) +$$

$$V_{held} = L * \sqrt{P} * (1 + \tau) - L * \left(\sqrt{P_a} + \frac{\tau * P}{\sqrt{P_b}} \right)$$

The V3 impermanent loss is thus after a quick computation :

$$\frac{V_{arbitraged} - V_{held}}{V_{held}} = IL_{v2}(\tau) * \frac{1}{1 - \frac{\sqrt{\frac{P_a}{P}} + \tau * \sqrt{\frac{P}{P_b}}}{1 + \tau}} = IL_{P_a, P_b}(\tau) \quad (54)$$

where $IL_{v2}(\tau) = \frac{2 * \sqrt{\tau}}{1 + \tau} - 1 = IL_{v2}(\tau)$ is the standard uniswap V2 impermanent loss for the range $[0, +\infty[$.

In the case $P_a = P_b = P$, then the impermanent loss will be 0.

$$\lim_{P_a \rightarrow 0, P_b \rightarrow +\infty} IL_{P_a, P_b}(\tau) = IL(\tau) \quad (55)$$

When the liquidity bounds are pushed to $[0, +\infty[$, the v3 impermanent loss goes back to the original v2 one.

Finally, setting τ to 1, we do get 0 since there should not be any impermanent loss in any scenario, v2 or v3.

1) *V3 Impermanent loss is larger:* If we take the simple assumption $p_a = \frac{P}{n}$ and $p_b = n * P$, we get the following loss :

$$IL_{P_a, P_b}(\tau) = IL(\tau) * \frac{1}{1 - \frac{1}{\sqrt{n}}} \quad (56)$$

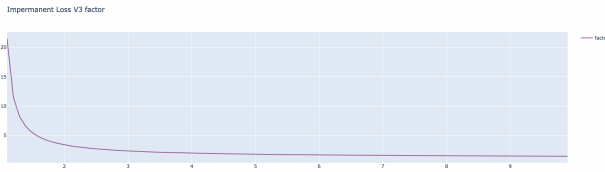


Fig. 24. Impermanent loss V3 factor

Even if our liquidity range is big enough to accommodate prices doubling or halving, impermanent loss is nearly 4 times higher than if we provided liquidity in the whole range of prices.

And that is excluding the impermanent loss associated with falling outside the concentrated liquidity range.

E. Uniswap V3 LP liquidity providing value

1) *Add/deletion of liquidity:* When adding or removing liquidity from a position, the amount of assets to add according

to the amount of added liquidity ΔL is given by the equivalent formula as 48:

$$\begin{cases} \begin{cases} \Delta X = 0 \\ \Delta Y = \Delta L * (\sqrt{P_b} - \sqrt{P_a}) \end{cases}, P \geq P_b \\ \begin{cases} \Delta X = \Delta L * \left(\frac{1}{\sqrt{P}} - \frac{1}{\sqrt{P_b}} \right) \\ \Delta Y = \Delta L * \left(\sqrt{P} - \sqrt{P_a} \right) \end{cases}, P_a \leq P \leq P_b \\ \begin{cases} \Delta X = \Delta L * \left(\frac{1}{\sqrt{P_a}} - \frac{1}{\sqrt{P_b}} \right) \\ \Delta Y = 0 \end{cases}, P \leq P_a \end{cases} \quad (57)$$

2) *Valuing a liquidity position:* The initial amount of tokens X_0 , Y_0 and L when entering a LP position with bounds P_a and P_b and initial price P_0 will follow 48.

You can deduct L value from 48 applied to X_0 , Y_0 and P_0 . The final amount of tokens X , Y and L when exiting a LP position with bounds P_a and P_b and current price P will also follow 48 for the previous L .

The liquidity position PL can be written as :

$$PnL = \frac{X * P + Y}{X_0 * P_0 + Y_0} - 1 \quad (58)$$