Active strategy for passive liquidity mining on Uniswap V3

Stefan Duprey
Blue Lama consulting
stefan.duprey@gmail.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 v3 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.**
However, the mathematical relationship between the liquidity of a position, the amount of assets in that position, and its price range becomes somewhat complex.
In a volatile market, finding the right bounds for an optimal trading liquidity concentration can be challenging though.
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 detect the perfect bounds.
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.
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.
Passive liquidity investing must be seen as a kind of mean reverting strategy where the money is made from price fluctuation inside a specified range.
Liquidity provider hopes the price will stay within definite maximal bounds so that the risk they take from price deviation is not too big.
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 an optimization problem to detect liquidity bounds in rangy market.

- **Maximize price inside your bounds to earn volume fees**
- **Tightest bound for capital efficiency**
- **Bounds re-balancing low frequency**

We here show that your optimal parameters triggering the state definition of a rangy market must be re calibrated with a definite frequency.
A whole framework to backtest the calibration period and the prediction period must be implemented.
A proper analysis of the rebalancing cost must also be done.
When our algorithm sees a trend in the market and does not define the market as "rangy", one should not bring liquidity as it mainly is a mean reverting strategy which won't perform in bullish/bearish market:
• In a bullish market, one should only possess full long the risky asset and not provide liquidity against a stable coin in Uniswap V3. If the asset consensus comes from a proof of stake, one should stake it. Otherwise, one should borrow on lending protocol to generate yield while still being long on the asset. • In a bearish market, the task is a bit harder. In the best case, you short the risky asset in dydx or perp.
We also propose an optimal liquidity profile allocation inside the optimal bounds.

I. Understanding the risk of hyperbolic dexes

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.

$$X * Y = cte = X_{t0} * Y_{t0} \tag{1}$$

The constant is fixed by the initially brought amount at instant $t0$. The risky asset price X in stable coins Y is directly linked to the pool reserve:

$$P_{X \text{ in } Y} = \frac{Y}{X} = P_X \tag{2}$$

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

$$P_{Y \text{ in } X} = \frac{X}{Y} = P_Y \tag{3}$$

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:

$$L = \sqrt{X * Y} \tag{4}$$

$$X = \frac{L}{\sqrt{P_X}} \tag{5}$$

$$Y = L * \sqrt{P_X} \tag{6}$$

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 arbitraged and a second one where the pool has been arbitraged.

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

$$P_{X_{t_2}} = \frac{Y_{t_2}}{X_{t_2}} = P_{consensus} \tag{7}$$

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

$$P_{X_{t_1}} = \frac{Y_{t_1}}{X_{t_1}} \neq P_{consensus} \tag{8}$$

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

$$V_{t_1} = X_{t_1} * P_{consensus} + Y_{t_1} = X_{t_1} * P_{X_{t_2}} + Y_{t_1} \tag{9}$$

$$V_{t_2} = X_{t_2} * P_{consensus} + Y_{t_2} = X_{t_2} * P_{X_{t_2}} + Y_{t_2} \tag{10}$$

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

$$\frac{v_{t_2} - v_{t_1}}{v_{t_1}} = \frac{2 * \sqrt{\frac{P_{X_{t_2}}}{P_{X_{t_1}}}}}{1 + \frac{P_{X_{t_2}}}{P_{X_{t_1}}}} - 1 = \frac{2 * \sqrt{P_{X_{t_1}} P_{X_{t_2}}} - (P_{X_{t_1}} + P_{X_{t_2}})}{P_{X_{t_1}} + P_{X_{t_2}}} \tag{11}$$

This function is symmetrical in P_{X_1} the price of the non-arbitraged pool at instant $t1$ and $P_{consensus} = P_{X_2}$ the consensus price at $t2$.
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 :

$$\frac{v_{t_2} - v_{t_1}}{v_{t_1}} = \frac{2 * \sqrt{\tau}}{1 + \tau} - 1 = IL_{v2}(\tau) \tag{12}$$

- 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

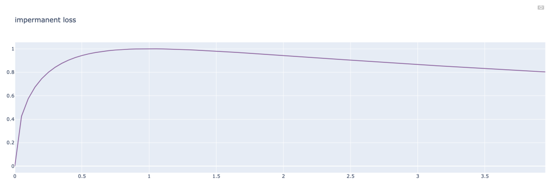

Fig. 1. Impermanent loss versus price ratio

- 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
- 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 $P_a <= P_b$ 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.

$$P = P_X = P_{X \text{ in } Y} = \frac{Y}{X} \tag{13}$$

With that notation, it comes :

$$L = \sqrt{X * Y} \tag{14}$$

$$X = \frac{L}{\sqrt{P}} \tag{15}$$

$$Y = L * \sqrt{P} \tag{16}$$

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

$$X_{virtual} * Y_{virtual} = L^2 \tag{17}$$

And the reserves evolution are dictated by

$$\Delta\sqrt{P} = \frac{\Delta Y}{L} \tag{18}$$

$$\Delta\frac{1}{\sqrt{P}} = \frac{\Delta X}{L} \tag{19}$$

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 :

$$X_b = \frac{L}{\sqrt{P_b}} \tag{20}$$

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

$$Y_a = L * \sqrt{P_a} \tag{21}$$

$$\begin{cases} X_{virtual} = X_{real} + \frac{L}{\sqrt{P_b}} \\ Y_{virtual} = Y_{real} + L * \sqrt{P_a} \end{cases} \tag{22}$$

The hyperbole equation thus becomes:

$$(X_{real} + \frac{L}{\sqrt{P_b}}) * (Y_{real} + L * \sqrt{P_a}) = L^2 \tag{23}$$

From now on, we will give up the real tag and call $X = X_{real}$ and $Y = Y_{real}$.
The solution is given by:
For $P \geq P_b$:

$$\begin{aligned} X &= 0 \\ Y &= L * (\sqrt{P_b} - \sqrt{P_a}) \end{aligned} \tag{24}$$

For $P \leq P_a$:

$$\begin{aligned} X &= L * \frac{\sqrt{P_b} - \sqrt{P_a}}{\sqrt{P_b} * \sqrt{P_a}} \\ Y &= 0 \end{aligned} \tag{25}$$

For $P_a \leq P \leq P_b$:

$$\begin{aligned} X &= X_{virtual} - \frac{L}{\sqrt{P_b}} = L * (\frac{1}{\sqrt{P}} - \frac{1}{\sqrt{P_b}}) = L * \frac{\sqrt{P_b} - \sqrt{P}}{\sqrt{P_b} * \sqrt{P}} \\ Y &= Y_{virtual} - L * \sqrt{P_a} \ = L * (\sqrt{P} - \sqrt{P_a}) \end{aligned} \tag{26}$$

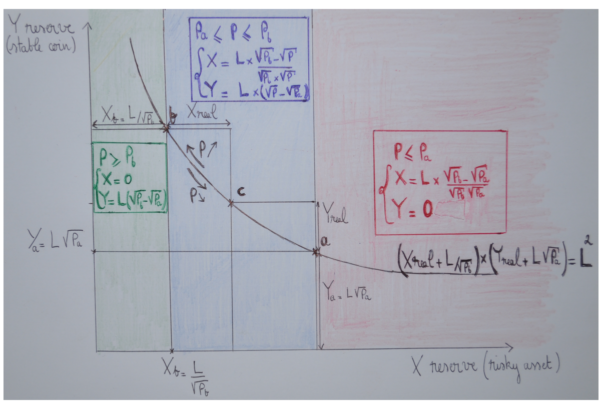

Fig. 2. Concentrated liquidity 3 regimes

D. Computing the impermanent loss

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

$$V = X * P + Y \tag{27}$$

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

$$V = 2 * L * \sqrt{P} - L * (\sqrt{P_a} + \frac{P}{\sqrt{P_b}}) \tag{28}$$

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

$$P_{consensus} = \tau * P \tag{29}$$

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

$$V_{arbitraged} = 2 * L * \sqrt{\tau * P} - L * (\sqrt{P_a} + \frac{\tau * P}{\sqrt{P_b}}) \tag{30}$$

$$V_{held} = X * P_{consensus} + Y \tag{31}$$

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

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

The V3 impermanent loss is thus after a quick computation :

$$\frac{V_{arbitraged} - V_{held}}{V_{held}} = IL_{v2}(\tau) * \frac{1}{1 - \frac{\sqrt{\frac{P_a}{P}} + \tau * \sqrt{\frac{P}{P_b}}}{1 + \tau}} = IL_{P_a, P_b}(\tau) \tag{32}$$

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.

$$\lim_{P_a, P_b \to +\infty} IL_{P_a, P_b}(\tau) = IL(\tau) \tag{33}$$

E. A mean reverting strategy

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.
The simple conclusion here is that liquidity provisioning in Uniswap V2 is by essence a mean reverting strategy.
Liquidity provider hopes the price will stay within definite maximal bounds so that the risk they take from price deviation is not too big.
An algorithmic detection is therefore a must : one should discriminate markets as rangy, bullish or bearish.
Liquidity provisioning should be seen as a mean reverting strategy only applicable to rangy markets.

II. Detecting Rangy Market

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 $P_a <= P_b$ and the liquidity brought by the user will only account for that range allowing a huge gain in capital efficiency.

A. Bollinger Bands

Bollinger Bands are a type of price envelopes plotted at a standard deviation level above and below a simple moving average of the price. Because the distance of the bands is based on standard deviation, they adjust to volatility swings in the underlying price.
Bollinger Bands use 2 parameters, the 'Period' parameter for the moving average and StDev or the number of standard deviations.

The default values are 20 for period, and 2 for standard deviations, although you may customize the combinations.
Bollinger bands help determine whether prices are high or low on a relative basis. They are used in pairs, both upper and lower bands and in conjunction with a moving average. Further, the pair of bands is not intended to be used on its own. Use the pair to confirm signals given with other indicators.

B. Defining rangy market from the Bollinger Bands

We choose to categorize markets as rangy as soon as the Bollinger Bands flatten, meaning the absolute value of the slopes.
of both the upper and lower Bollinger Bands are below a specific threshold (which is obtained by looking at the past distribution quantiles).
When a rangy market is detected, we freeze the upper and lower Bollinger Bands levels defining the beginning of a rangy market and wait till the absolute value of the slopes trespass the threshold, meaning the end of the rangy market.
The past period 'slopePeriod' used for the slope calculation and the thresholds 'slopeThreshold' are two new parameters for our "rangy market" model.

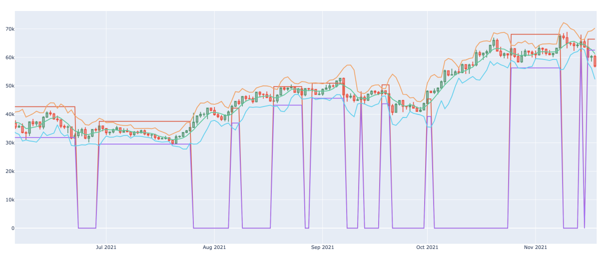

Fig. 3. Example of rangy market detection

1) Getting the optimal parameters: 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 detect the perfect bounds.
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.
So liquidity provisioning has to be actively monitored. The chosen bounds must be actively managed to encompass the price moves and not get traversed.
Passive liquidity investing must be seen as a kind of mean reverting strategy where the money is made from price fluctuation inside a specified range.
Of course rebalancing the bounds is costly and must happen not too often.
The chosen bounds must also fit perfectly the price envelope for the optimal capital efficiency of your liquidity provisioning.
So in other words, we end up with the following constrained optimization problem :

- Maximize price inside your bounds to earn volume fees
- Tightest bound for capital efficiency
- Bounds rebalancing low frequency

That optimization problem is a complex one, which has to be posed properly. The rebalancing and price fitting constraints can be added as a penalty in the objective functions through Lagrange multiplier.

$$Period^*, StDev^*, slopePeriod^*, slopeThreshold^* = \sup_{Period, StDev, slopePeriod, slopeThreshold} \begin{cases} \sum_d rangy(d) \\ -C * discrepancy \\ -D * rebalancing \end{cases} \tag{34}$$

$$rangy(d) = \mathbf{1}_{P_{high}(d) < BollUpper(Period, StDev)(d)} * \mathbf{1}_{P_{low}(d) > BollLower(Period, StDev)(d)} \tag{35}$$

III. Complexity of the Optimization

The optimization of the previous section is a complex problem. We will here cover a few problems arising.

A. Risk aversion

As in Markowitz portfolio optimization, the Lagrange multiplier can be seen as risk aversion parameters and will be defining characteristics/parameters of our strategy.

B. Optimization per stages

The optimal parameters will depend on the time range where you fit them. You won't find the same parameters if you restrict the time range to the current bull market since the beginning of 2021 or if you look for optimal parameters during the 2018-2019 crypto winter.

1) Crypto winter parameters: We here present rangy markets detection from an exhaustive search over the period 2018-2020. The optimal parameters found are :

- bb periods = [15]
- stdNbrs = [3]
- slope windows = [5]
- slope thresholds = [190]

We can see that those parameters do a decent job during the years 2018-2020 where they have been optimized. Nevertheless during the actual following bull run, they are a bit too conservative and detect almost no rangy market phase.

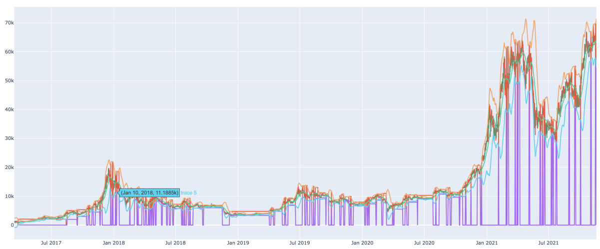

Fig. 4. Crypto winter parameters overall view

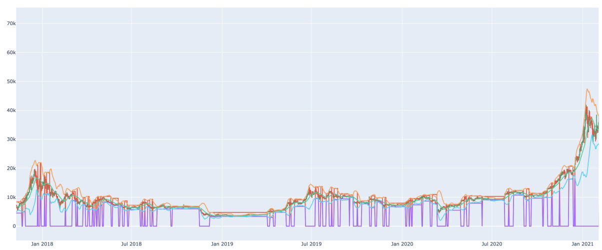

Fig. 5. Crypto winter parameters winter zoom

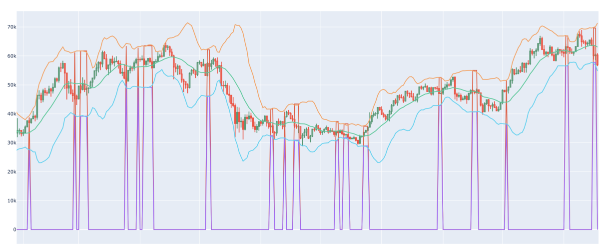

Fig. 6. Crypto winter parameters bull zoom 2020-2021

2) Bull run parameters: We here present rangy markets detection from an exhaustive search over the period 2020-2021. The optimal parameters found are :

- bb periods = [5]
- stdNbrs = [3]
- slope windows = [5]
- slope thresholds = [590]

We can notice that the bollinger period is smaller than the winter one as the slope threshold thus triggering the definition of a rangy market more easily.
This is why the bull market We see that those parameters totally miss the good ranges between 2018 and 2020 as they have been optimized afterwards during 2020-2021.
Nevertheless, they catch a bigger part of the rangy markets since 2021 still allowing to profit from passive liquidity mining strategies even in bullish markets.

3) Backtesting, overfitting and transaction cost: The clear demonstration before is that your optimal parameters triggering the state definition of a rangy market must be recalibrated every once in a while.

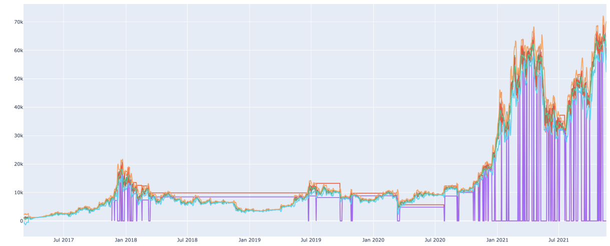

Fig. 7. Bull run parameters

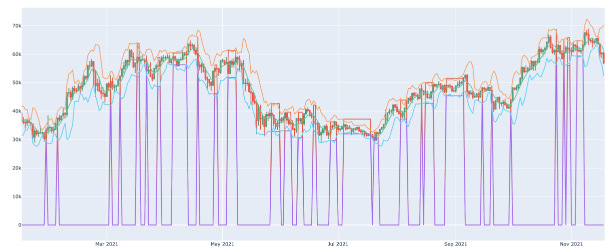

Fig. 8. Bull run parameters bull zoom 2020-2021

A whole framework to backtest the calibration period and the prediction period must be implemented.
A proper analysis of the rebalancing cost must also be done.

IV. Optimizing liquidity profile inside rangy market bounds

A. Tick definition

Tick mathematical understanding is required to interpret both the values provided by the Uniswap v3 API and the data indexed in the Uniswap v3. Uniswap v3 maps the continuous space of all possible prices to a discrete subset indexed by ticks. A tick has unique relation with price, defined by the tick base parameter, which is equal to 1.0001 in Uniswap (a one basis point deviation in price). The price corresponding to the i-th tick is :

$$p(i) = 1.0001^i \tag{35}$$

However pools actually track ticks at every square root price that is an integer power of $\sqrt{1.0001}$. The tick i is thus defined :

$$\sqrt{p(i)} = \sqrt{1.0001}^i \tag{35}$$

$$i_c = \left[\log_{\sqrt{1.0001}} \sqrt{P}\right] \tag{35}$$

B. Customized liquidity bringing

We here follow the same idea as in 'https://uniswaptimizer.com/' and try to detect an optimal profile per tick for liquidity provisioning in Uniswap V3 inside a specific range.

C. Optimal profile

Find the average daily volume of a pool averaged over a week
Calculate the liquidity of a potential position for alreadu fixed start and end points.
Calculate a distribution of the existing liquidity overlapping with each bucket.
This is similar to the views in both the flipside calculator and the liquidity view of the Uniswap analytics dashboard.
Use the normal distribution of the price pair (using one week of daily price movement to calculate volatility = standard deviation) to calculate the probability of the price being in each bucket. This of course makes the incorrect assumption that the past volatility will reflect the future, but hey, we gotta start somewhere. We can sum up all the buckets of the total start/end range to get an expected value for total 'liquidity coverage'.
Pulling everything together the calculation comes out to:

- Daily Fees = Pool Volume * Pool Fee % * Liquidity Coverage Expected Value
- APY = (Daily Fees / 1000) * 365

Finally choose the highest APY position for each pool

D. Add/deletion of liquidity

$$\Delta Y = \begin{cases} 0 & i_c < i_l \\ \Delta L * (\sqrt{P} - \sqrt{P(i_l)}) & i_l \leq i_c < i_u \\ \Delta L * (\sqrt{P(i_u)} - \sqrt{P(i_l)}) & i_c \geq i_u \end{cases} \tag{35}$$

$$\Delta X = \begin{cases} L * (\frac{1}{\sqrt{P(i_l)}} - \frac{1}{\sqrt{P(i_u)}}) & i_c < i_l \\ L * (\frac{1}{\sqrt{P}} - \frac{1}{\sqrt{P(i_u)}}) & i_l \leq i_c < i_u \\ 0 & i_c \geq i_u \end{cases} \tag{35}$$

V. Dealing with the "non-rangy" phases

When our algorithm sees a trend in the market and does not define the market as "rangy", one should not bring liquidity as it mainly is a mean reverting strategy which won't perform in bullish/bearish market.
• In a bullish market, one should only possess full long the risky asset and not provide liquidity against a stable coin in Uniswap V3. If the asset consensus comes from a proof of stake, one should stake it. Otherwise, one should borrow on lending protocol to generate yield while still being long on the asset.
• In a bearish market, the task is a bit harder. In the best case, you short the risky asset in dydx or perp.

VI. Conclusion

https://arxiv.org/pdf/2106.14404.pdf
https://uniswapv3.flipsidecrypto.com/
https://uniswaptimizer.com/ https://app.flipsidecrypto.com/velocity/queries/efed0457-5edc-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