Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Providing Liquidity in Uniswap V3

Bachelor's Thesis

Yann Huynh
yhuynh@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Robin Fritsch
Prof. Dr. Roger Wattenhofer

March 20, 2022

Acknowledgements

I want to thank my supervisor Robin Fritsch for his continuous feedback, guidance, and insight about the topic. Furthermore, I would like to thank Prof Dr. Roger Wattenhofer for allowing me to do this project and the Disco group for providing me computational resources to complete my thesis.

I would also like to thank Moussab Katouh for proofreading my thesis and Jan Stauffer for helping me with latex graphics as in Figure 4.2.

Abstract

One of the largest DEX Uniswap released their version v3 in 2021 and introduced the concept of concentrated liquidity. Now liquidity providers in Uniswap v3 need to choose a range in which they want to provide liquidity. We built a robust and precise backtester based on the original Uniswap v3 smart contract and used it to test eleven different strategies for liquidity providers. We analyzed those strategies on the USDC-ETH pool with 0.05% transaction fee. We found strategies that performed really well when Ethereum was going up, but most of these strategies did not perform so well when Ethereum was going down. We also analyzed the delta of liquidity provision strategies.

Contents

Acknowledgements **i**

Abstract **ii**

1 Introduction **1**

1.1 Related Work 2

2 Background **3**

2.1 Decentralized Exchanges 3

2.2 Constant Product Market Makers 3

2.3 Concentrated Liquidity 4

2.3.1 Range Orders 6

2.4 Uniswap V3 Implementation Details 7

2.4.1 Ticks and Ranges 7

2.4.2 Global State 7

2.4.3 Tick-Indexed State 9

2.4.4 Position-Indexed State 9

3 Methodology **11**

3.1 Data 11

3.1.1 Pools 11

3.2 USDC 11

3.3 Backtester 12

3.3.1 Analysis 12

3.4 Metrics 13

3.4.1 Return On Investment 13

3.4.2 Maximum Drawdown 13

3.4.3 Volatility 13

3.4.4 Sharpe Ratio 14
3.4.5 Downside Deviation 14
3.4.6 Sortino Ratio 15
3.4.7 Value at Risk 15
3.4.8 Delta 15

4 Strategies 16

4.1 No Provision 16
4.2 Uniswap V2 16
4.3 Constant Interval 17
4.4 Interval Around the Current Price 17
4.5 Two Intervals Around the Current Price 18
4.6 Fill Up 18
4.7 Swap 19
4.8 Range Order 19
4.9 Moving Average 20
4.10 Volatility sized Interval 20
4.11 Bollinger Bands 20

5 Results 21

5.1 Buying Ethereum 21
5.2 Strategies 21
5.2.1 No Provision 22
5.2.2 V2 Strategy 22
5.2.3 Constant Interval 24
5.2.4 Interval Around the Current Price 25
5.2.5 Two Intervals the Around Current Price 27
5.2.6 Fill Up 28
5.2.7 Swap 30
5.2.8 Range Order 31
5.2.9 Moving Average 33
5.2.10 Volatility Sized Interval 34
5.2.11 Bollinger Bands 36

5.3 Detailed Analysis of the Interval Around the Current Price Strategy 38
5.3.1 Monthly Analysis 40
5.4 Detailed Analysis of the Fill up Strategy 41
5.5 Delta 43
5.5.1 Constant Interval 43
5.5.2 Interval Around the Current Price 44
5.5.3 Range Order 45
5.5.4 Bollinger Bands 45
5.5.5 Comparisons Between Strategies 46

6 Conclusion and Future Work **47**

Bibliography **48**

Chapter 1

Introduction

The concept of blockchain and the first decentralized cryptocurrency has been introduced by Nakamoto with Bitcoin [1]. Later blockchains with smart contracts, most notably Ethereum [2] followed. Due to smart contracts, decentralized finance or DeFi emerged on Ethereum. DeFi is a financial technology that eliminates the need for traditional financial intermediaries like banks. Various DeFi protocols provide different services to users. These protocols allow users to take out loans, get insurance, or exchange tokens without intermediaries involved.

Protocols that allow users to exchange tokens on-chain are called Decentralized Exchanges (DEX). These exchanges allow users to trade without an intermediary. Traders do not have to deposit their funds in a centralized exchange (CEX) and therefore give up custody of their funds. They can instead exchange tokens directly on-chain with smart contracts. Most of these DEXs use an automated market-making algorithm rather than a traditional order book. The orders are executed on a liquidity pool. Liquidity providers (LP) deposit that liquidity in a pool, and traders trade against that. As a reward for providing this liquidity, LPs get a fee for each swap.

In Uniswap v3 [3], LPs must also choose price ranges for a particular pool in which they want to provide their liquidity. The LP's fee depends on the selected price range and the current price. Providing liquidity also comes with a risk. An LP will suffer impermanent loss if the price of a pair of tokens is different. The range the LP has chosen determines how big that impermanent loss is.

We investigate different strategies to choose these price ranges and how impermanent loss and fees affect the returns of LPs. For that investigation, we backtested these strategies on past data.

1.1 Related Work

Max [4] introduced the passive rebalancing strategy for Uniswap v3 in a blog post. Gamma Strategies developed a strategy based on Bollinger Bands [5]. Clark derived the replicating portfolio and greeks for constant product market [6] and for constant product with bounded liquidity [7]. Neuder et al. [8] did a formal study on liquidity provision strategies on Uniswap v3 with a Markov model. However, they did not consider impermanent loss. Fritsch [9] analyzed different liquidity provision strategies for Uniswap v3. He used hourly data to analyze the strategies. We have improved the analysis by doing more exact calculations and adding more strategies. Mellow showed how to create different market making strategies on Uniswap v3. They also show how to build derivatives on-chain [10].

CHAPTER 2

Background

2.1 Decentralized Exchanges

On decentralized exchanges like Uniswap, Sushiswap, and Pancakeswap, traders can swap between different tokens on the DEX, which does not require traders to deposit any currency in the exchange, allowing traders always to have custody of their assets. The swap happens atomically, and DEX never holds any of the trader's assets. Transactions in DEXs are executed on-chain. Therefore, the trade is only settled when the network has verified the transaction.

There are DEXs on various platforms such as Solana, Avalanche, but Ethereum is the most popular. On Ethereum, any token that fulfills the ERC-20 standard can be traded on a DEX. Such a token can represent a lot of things, like a lottery ticket, shares in a company, or a US dollar [11]. Users can trade almost any asset on DEXs because tokens do not need to be listed by a central authority. Instead, any user can open a new liquidity pool for a token pair. Furthermore, DEXs are not subject to governments and regulatory entities. Users are free to trade and are never required to do KYC.

2.2 Constant Product Market Makers

In contrast to CEXs, most DEXs do not use a traditional order book, but they use an automated market maker (AMM). Most DEXs like Sushiswap and Uniswap, which is currently the largest DEX and has a total locked value of over $8 billion and a daily volume of $1-2 billion [12, 13], use a form of constant product market maker (CPMM).

In Uniswap v2 [14], anyone can create a liquidity pool for an arbitrary token pair. A liquidity provider will deposit both tokens into the pool to bootstrap that pool. Once the pool is bootstrapped, a liquidity provider must always deposit both tokens in the same ratio as the pool's current state. Traders swap tokens with the pool's liquidity and pay a fee. This fee will then be distributed pro-rata among all liquidity providers. The trading fee of Uniswap v2 is 0.3%.

A CPMM like Uniswap v2 ensures that the product of the two reserves stays constant. Consider a pool of the token pair (X, Y). Assume the reserve for token X is x and the reserve for token Y is y. Then the product $x \cdot y = k$ is constant if the pool has no fees. Consider a trader who wants to swap y of token Y to token X. With fees, the following product is constant, $k = (x - x_{out}) \cdot (y + (1-f)y_{in})$, where f is the transaction fee. After the swap, the new reserves will be:

$$y' = y + (1-f)y_{in} \tag{2.1}$$

$$x' = \frac{k}{y'} = \frac{k}{y + (1-f)y_{in}} \tag{2.2}$$

So the amount of token B the trader gets is t_B:

$$x_{out} = x - x' = x - \frac{k}{y + (1-f)y_{in}} \tag{2.3}$$

[15] The reserves can never go to zero, as we can see in Equation 2.2, $x' = 0 \iff y_{in} \to \infty$. Such a trade would have infinite costs. From Equation 2.3 we see that big orders have a bigger price impact and are therefore more expensive than small orders. The price of an asset is only determined by the ratio of the reserves.

$$P = \frac{x}{y} \tag{2.4}$$

Arbitrageurs will keep the price close to prices on other exchanges.

2.3 Concentrated Liquidity

With Uniswap v3, the concept of concentrated liquidity got introduced. In earlier versions of Uniswap, the liquidity providers had to support trading on the entire price range $(0, \infty)$. The liquidity was uniformly distributed along:

$$k = x \cdot y \tag{2.5}$$

This distribution is not capital efficient because assets only trade in a small subset of $(0, \infty)$, much of the liquidity in a pool is never used.

It makes sense to allow LPs to provide liquidity in a smaller price range than $(0, \infty)$. We call a finite price range $[p_a, p_b]$ a position. A position only needs to support trading within its range. It will act like a CPMM with larger reserves within its range. These reserves will be called virtual reserves.

When the price moves down, it means that the reserves of Y are shrinking, and analogously, the reserves of X are shrinking when the price moves up. In 2.1

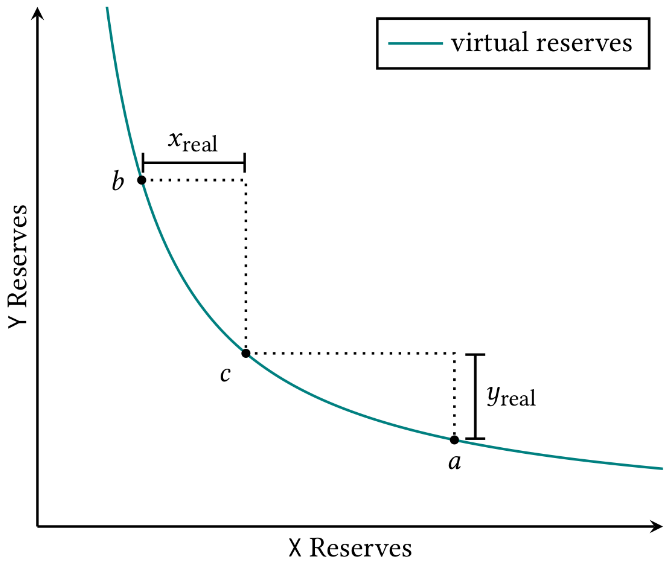

Figure 2.1: Virtual Liquidity [3]

we see that, for a price range $[p_b, p_a]$ and a current price p_c with $p_c \in [p_b, p_a]$, we would need to provide x_{real} of X and y_{real} of Y.

When the price moves out of the range, $p_c \notin [p_b, p_a]$, the reserves of that position are entirely made up of one asset. The position is no longer active, and it does not earn any fees. When the price moves back in $p_c \in [p_b, p_a]$, the position will become active and earn fees again.

The amount of liquidity needed for a position is measured by $L = \sqrt{k}$. The real reserves of a position are:

$$L^2 = (x + \frac{L}{\sqrt{p_b}}) \cdot (y + L\sqrt{p_a}) \tag{2.6}$$

The curve of the real reserves (Equation 2.6) is a translation of (Equation 2.5). This relation can be seen in Figure 2.2.

LPs can create as many positions as they want, giving them the possibility to approximate any distribution of liquidity. The concept of concentrated liquidity allows the market to decide in which price range liquidity is needed. Concentrated liquidity is a mix of a classical order book and a CPMM that only allows LPs to provide the whole price range $(0, \infty)$.

[3]

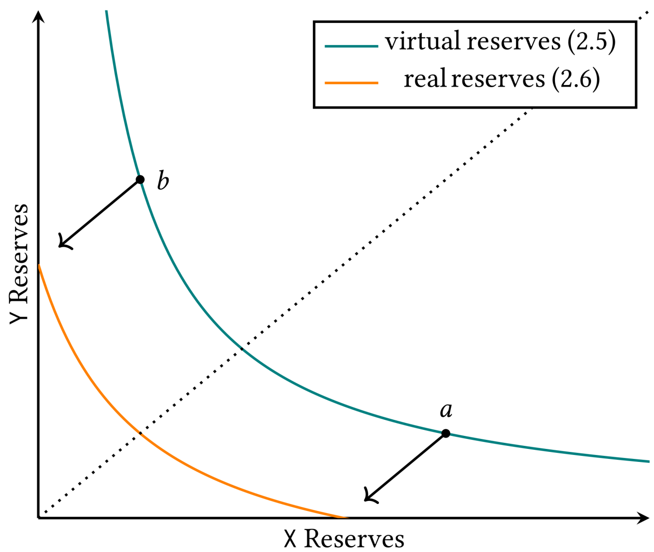

Figure 2.2: Real Reserves [3]

2.3.1 Range Orders

We can approximate limit orders by providing liquidity in a small interval. The difference between a range order and a traditional limit order is that the range order has a minimum interval width, and while the price is in that interval, the order is partially executed. Another difference is that we need to withdraw if the price crosses the range. If we do not withdraw it and the price crosses back again, the position will be traded back and reverse the trade. Since range orders are just a regular liquidity provision, we do not have to pay swap fees. On the contrary, we even get fees.[3]

Example 2.1. Assume the current price p_c of ETH is at 100 USDC, and we want to sell ETH. We set a range order at $[150, 151]$. When the price moves above 151, we have sold all of our ETH for USDC.

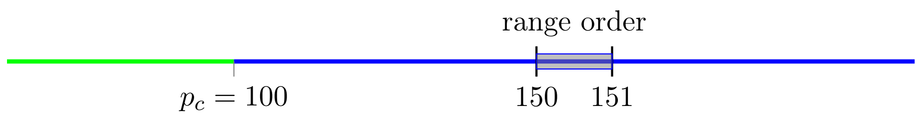

Figure 2.3: Range Order before Execution

Figure 2.4: Range Order after Execution

2.4 Uniswap V3 Implementation Details

Implementing concentrated liquidity in a gas-efficient manner is not easy. Uniswap v3 has a few interesting implementation details. These details are described in [3] and also implemented in our backtester to ensure that we exactly replicate the smart contract. The smart contract of Uniswap v3 tries to optimize for swaps, since swaps are by far the most common type of transaction. We will not consider any protocol fees because they are zero at the moment.

2.4.1 Ticks and Ranges

The space of possible price ranges is divided into discrete ticks. A position in Uniswap v3 is made up of a lower tick i_l and an upper tick i_u. The price of the tokens is always expressed as the price of token0 in terms of token1. The choice of which asset is token0 or token1 is arbitrary and does not change the logic of the contract, besides rounding errors.

There is a tick for every price, which is a power of 1.0001. The ticks are indexed by an integer i, and the price of each index is:

$$p(i) = 1.0001^i \tag{2.7}$$

Therefore, each tick is 0.01% or one basis point away from its neighboring ticks. Not every tick can be used, but only certain ticks depending on the tick spacing. [3]

2.4.2 Global State

The important variables of the contract that are relevant to swaps and liquidity provision are:

Variable Name	Notation
liquidity	L
sqrtPrice	$\sqrt{P}$
tick	i_c
feeGrowthGlobalToken0	$f_{g,0}$
feeGrowthGlobalToken1	$f_{g,1}$

Table 2.1: Global State [3]

The pool does not track the current reserves, x and y but instead tracks L and $\sqrt{P}$. We can compute x and y by:

$$x = \frac{L}{\sqrt{P}} \tag{2.8}$$

$$y = L \cdot \sqrt{P} \tag{2.9}$$

The global state also contains the current tick i_c, which is given by:

$$i_c = \lfloor \log_{\sqrt{1.0001}} \sqrt{P} \rfloor \tag{2.10}$$

Furthermore, the contract tracks the two numbers feeGrowthGlobalToken0 ($f_{g,0}$) and feeGrowthGlobalToken1 ($f_{g,1}$). These numbers represent total fees earned per unit of virtual liquidity (L) over the entire contract history. [3]

Swap Within a Single Tick

Within a single Tick, the contracts act as a regular CPMM. Suppose the transaction fee is γ, and we swap Δy token1 for token0. Then feeGrowthGlobalToken1 will be incremented by:

$$\Delta f_{g,1} = y_{in} \cdot \gamma \tag{2.11}$$

The sqrtPrice will be increased by:

$$\Delta y = y_{in} \cdot (1 - \gamma) \tag{2.12}$$

$$\Delta \sqrt{P} = \frac{\Delta y}{L} \tag{2.13}$$

These formulas only work if we do not cross any tick. Otherwise, we can only use a portion of y_{in}, then cross ticks, as described in section 2.4.3, and then continue swapping in a single tick. [3]

2.4.3 Tick-Indexed State

Each tick tracks the following values:

Variable Name	**Notation**
liquidityNet	ΔL
liquidityGross	L_g
feeGrowthOutsideToken0	$f_{o,0}$
feeGrowthOutsideToken1	$f_{o,1}$

Table 2.2: Tick-Indexed State [3]

LiquidityNet ΔL is the amount of liquidity that should be removed or added when the tick is crossed. LiquidityGross is used to determine if a tick has an active position. We can not use ΔL, since $\Delta L = 0$ does not guarantee that the tick is not used.

FeeGrowthOutsideToken{0,1} are used to calculate the fees within a given range. We will omit the subscript for the rest of this section because the formulas are identical for both tokens. Let us define f_a as the fees earned above a tick i:

$$f_a(i) = \begin{cases} f_g - f_o(i) & i_c \geq i \\ f_o(i) & i_c < i \end{cases} \tag{2.14}$$

And analogously f_b as the fees below a tick i:

$$f_b(i) = \begin{cases} f_o(i) & i_c \geq i \\ f_g - f_o(i) & i_c < i \end{cases} \tag{2.15}$$

Then we can calculate the fees in the range f_r between a lower tick i_l and an upper tick i_u as follows:

$$f_r = f_g - f_b(i_l) - f_a(i_u) \tag{2.16}$$

Cross Ticks

Whenever we cross ticks, we need to update f_o. It will be updated as follows:

$$f_o(i) := f_g - f_o(i) \tag{2.17}$$

After the crossing, we can continue swapping, as described in section 2.4.2. [3]

2.4.4 Position-Indexed State

The contract has a mapping from (user, lower tick, upper tick) to a position struct, which tracks the following values:

Variable Name	Notation
liquidity	l
feeGrowthInsideToken0Last	$f_{r,0}(t_0)$
feeGrowthInsideToken1Last	$f_{r,1}(t_0)$

Table 2.3: Position-Indexed State State [3]

The liquidity (l) is the virtual liquidity of the position. The liquidity does not change as fees are earned. In contrast to Uniswap v2 where the fees are reinvested automatically and therefore increase the liquidity.

Set Position

The function setPosition takes three arguments: liquidityDelta δl, lowerTick i_l and upperTick i_r. First, it will calculate the uncollected fees (f_u) for both tokens. For that, it considers the change of f_r of $[i_l, i_r]$, between $t1$ and $t0$ and multiply it with the liquidity l:

$$f_u = l \cdot (f_r(t_1) - f_r(i_0)) \tag{2.18}$$

The fees will be payed out to the user and the function will update $f_{r,0}(t_0)$ and $f_{r,1}(t_0)$. Then the funciton updates l by adding δl. The following formulas calculate the amount the user needs to deposit (or receive if it is negative):

$$\Delta Y = \begin{cases} 0 & i_c < i_l \\ \Delta L \cdot (\sqrt{P} - \sqrt{p(i_l)}) & i_l \leq i_c < i_u \\ \Delta L \cdot (\sqrt{p(i_u)} - \sqrt{p(i_l)}) & i_c \geq i_u \end{cases} \tag{2.19}$$

$$\Delta X = \begin{cases} \Delta L \cdot (\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}}) & i_c < i_l \\ \Delta L \cdot (\frac{1}{\sqrt{P}} - \frac{1}{\sqrt{p(i_u)}}) & i_l \leq i_c < i_u \\ 0 & i_c \geq i_u \end{cases} \tag{2.20}$$

[3]

Chapter 3

Methodology

To evaluate our strategies, we use actual data from Uniswap pools. We then backtested the strategies with our backtester, which implements the core part of the Uniswap v3 smart contract.

3.1 Data

Uniswap v3 transaction data can be assessed using The Graph's Subgraph of Uniswap v3 [16]. Unfortunately, the data on The Graph was not correctly indexed. Therefore, we got the data directly out of the Ethereum blockchain. That data came from the Etherscan API [17]. This data had to be decoded using the contract ABI of Uniswap v3.

3.1.1 Pools

The analysis of the strategies was done on the Uniswap v3 ETH-USDC 0.05% pool, which was initialized on May 5, 2021. The in-sample period was from June 4, 2021, until November 20, 2021 and has a length of 169 days. The out-of-sample period is from November 20, 2021, until February 28, 2022 and has a length of 169 days. We chose to start backtesting one month after the initialization of the pool so that there is enough liquidity and volume.

3.2 USDC

We will assume for the whole thesis that 1USDC = 1$. This assumption is reasonable because each USDC is backed by a dollar and this backing is regularly audited [18].

3.3 Backtester

To evaluate our strategies, we implemented our backtesting tool. Our initial approach was running a local EVM using Hardhat [19] and deploying the original Uniswap v3 core smart contract [20]. However, it turned out that running a local blockchain is way too slow. That is why we decided to implement the logic of the contract in Go. While we built the backtester from scratch to leverage 256-bit integer calculations, we referenced code of similar implementations [21, 22].

Our backtester runs all relevant historical transactions (Mint, Burn, Swap, and Flash), and then it runs the transactions of our strategy on top of that. The backtester replicates the logic of the smart contract and comes to the same result as the original contract. We will not try to differentiate which portion of the end value comes from the value of the liquidity provision and which part comes from the fees because the backtester is automatically compounding the fees.

Our Go backtester achieved a 50x speed improvement due to various optimizations compared to a backtester based on the Uniswap v3 SDK [21]. The backtester with a local EVM needs several hours to complete one run of all transactions, whereas our Go backtester only needs about 1 second. The most important optimizations are caching complex calculations, using 256bit integers, and changing from Javascript to Go. Our backtester is also able to test different parameters in parallel.

The backtester uses the price of the pool (Equation 2.4) to calculate how much USDC our ETH is worth, which means that we assumed that the price on Uniswap v3 is reflective of the market price of Ethereum overall. We also used the assumption from section 3.2 to convert the USDC to dollar.

3.3.1 Analysis

All strategies start with an equal amount of both assets, and the total value of the assets will be $2. We chose such a small amount to minimize our strategies' impact on the pool's liquidity. As already mentioned in section 3.1.1 all strategies start one month after the initialization of the pool. Moreover, we did not consider any cost of gas fee when doing our analysis. We did not consider the gas cost because the cost of gas is constant and would be negligible with more capital. Depending on the strategy, there will be specific parameters. Strategies and their parameters will be discussed in chapter 4.

3.4 Metrics

We used several metrics to measure the performance of our strategies.

3.4.1 Return On Investment

The most important performance metric is return on investment, or ROI. It is defined by:

$$ROI = r = \frac{\text{end amount} - \text{start amount}}{\text{end amount}} \tag{3.1}$$

This return needs to be annualized:

$$r_a = (1 + r)^{\frac{365}{\text{days}}} - 1 \tag{3.2}$$

[23]

3.4.2 Maximum Drawdown

The drawdown (DD) of a portfolio is the loss from the peak. It is defined by:

$$DD = \frac{\text{value} - \text{peak value}}{\text{peak value}} \tag{3.3}$$

The maximum drawdown (MDD) is the maximum of all drawdowns. In our simulation, we take hourly Snapshots of the value of our positions and then calculate the maximum drawdown.

$$MDD = \max(DD) \tag{3.4}$$

[23]

3.4.3 Volatility

To estimate the volatility of our strategies, we take snapshots at fixed intervals (hourly, daily, and weekly). Define:

$n + 1$: Number of snapshots

V_i: Value of all positions at the end of the ith interval, for i from 0 to n

τ: Length of time interval in years.

Let

$$u_i = \ln\left(\frac{V_i}{V_{i-1}}\right), \text{ for } i \text{ from 1 to } n$$

be the change of V_i.
Let

$$\bar{u} = \frac{1}{n}\sum_{i=1}^{n} u_i$$

be the average of u_i.
The unbiased sample variance u_i is:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(u_i - \bar{u})^2} \tag{3.5}$$

From this we can estimate the volatility σ by $\hat{\sigma}$, which is defined as:

$$\hat{\sigma} = \frac{s}{\sqrt{\tau}} \tag{3.6}$$

[24]

3.4.4 Sharpe Ratio

The Sharpe ratio measures the ratio of the risk premium to the volatility.

$$\text{Sharpe Ratio} = \frac{r - r_f}{\sigma} \tag{3.7}$$

r is the return and r_f the risk-free rate. Usually, the US Treasury bill is taken as the risk-free rate. However, considering this thesis is about crypto, we will choose Aave's deposit rate on USDC.

3.4.5 Downside Deviation

The downside deviation is similar to the volatility, but it only looks at returns that fell below the minimum acceptable return (MAR). We will choose $MAR = 0$. The target downside deviation (TDD) is defined by:

$$TDD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\min(0, u_i - MAR)} \tag{3.8}$$

[25] This also needs to be scaled by the length of the interval. The downside volatility σ_d is estimated by $\hat{\sigma_d}$, which is defined by:

$$\hat{\sigma_d} = \frac{TDD}{\sqrt{\tau}} \tag{3.9}$$

3.4.6 Sortino Ratio

The Sortino ratio is similar to the Sharpe ratio but only considers the downside deviation. [25]

$$\text{Sortino ratio} = \frac{r - r_f}{\sigma_d} \tag{3.10}$$

We will also use Aave's rate as the risk-free rate here.

3.4.7 Value at Risk

The Value at Risk or VaR is the threshold loss value for a given portfolio, time, and a probability p. To calculate VaR, we take snapshots at fixed Intervals, sort them by the returns, and take the value of the p-th percentile.

3.4.8 Delta

The delta (Δ) of the liquidity postions is the ratio of change in value of the positions to the change in value of the underlying token. In our case, the underlying token is Ethereum.

Example 3.1. Let us assume the total value of our postitions increased from 2$ to 3$ and the price of Ethereum increased from 2000$ to 4000$. To compare the delta of liquidity positions to the delta of options we will scale it with the position size. The delta would then be:

$$\Delta = \frac{3 - 2}{4000 - 2000} \cdot \frac{2000}{2} = \frac{1}{2000} \cdot 1000 = 0.5 \tag{3.11}$$

[24] A delta of zero means that the liquidity is not subject to price change in the underlying asset. Delta one means that the change in the value of the liquidity position is identical to the change in the underlying asset.

Chapter 4

Strategies

We will backtest a few strategies to determine the best strategy for an active liquidity provider. For all strategies, we will test out all reasonable parameters. Furthermore, all our strategies do not try to predict the price of the tokens in the future. If the strategy is dynamic, we will rebalance our position every 2 hours, 6 hours, one day, seven days, and 30 days.

4.1 No Provision

In this strategy, half of the portfolio is in token zero, and the other half is in token one. The assets will not be deposited in any pool. This simple strategy is a good benchmark because it shows how much the value of the tokens increased. Since no liquidity was provided, there is no impermanent loss of earned fees. This strategy will be our reference strategy, which we will take as a baseline.

4.2 Uniswap V2

For this strategy, we will provide the liquidity just as in Uniswap v2. We will provide liquidity in the interval $[\text{minTick}, \text{maxTick}]$.

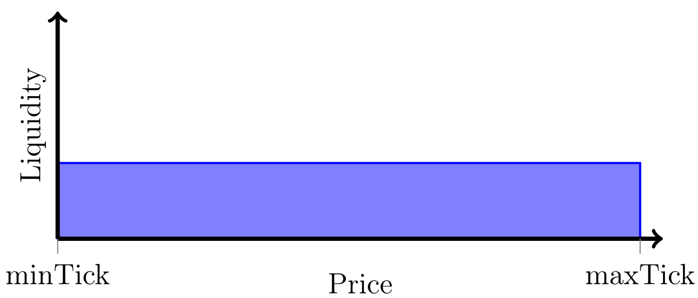

Figure 4.1: Uniswap v2 Strategy

4.3 Constant Interval

We will choose a fixed parameter $a > 0$ for this strategy. In the beginning, we will provide liquidity in a symmetric interval around the current price. This interval will never be adjusted. More specifically, we will choose the interval $[p-a, p+a]$, where p is the price at the beginning.

For the USDC-ETH pool, we will choose $a \in [10, 40000]$, which means that the size is between 10 and 40000 basis points, or 0.1% and 400%. It does not make sense to choose another a because, at any time, the price is always in $[p-a, p+a]$.

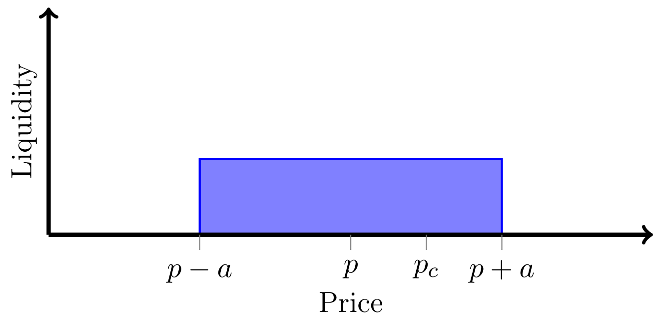

Figure 4.2: Constant Interval Strategy

4.4 Interval Around the Current Price

This strategy will provide liquidity around the current price in a fixed interval. We will rebalance our position every update interval as discussed in the beginning of chapter 4. Let p_c be the current price and $a \in [10, 40000]$, then we will set our position to $[p_c - a, p_c + a]$ at every update interval.

We will fill the interval as much as possible. However, it is usually not possible to use up both tokens, so the remaining token will be left outside the pool.

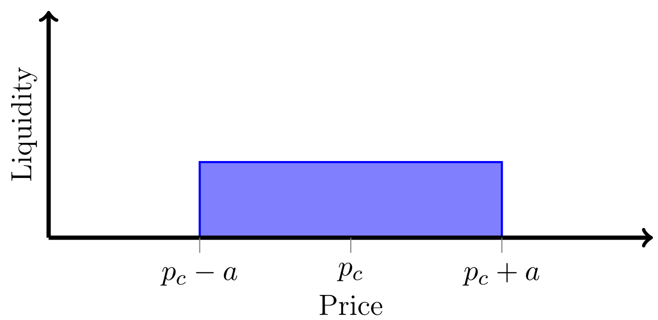

Figure 4.3: Interval Around the Current Price Strategy

Example 4.1. Let us consider a USDC-ETH pool. With the current price of one ETH being 100 USDC. Assume we have 1 ETH and 110 USDC. Then we will provide liquidity in the interval $[p_c - a, p_c + a]$ for some a. For that, we will use 1 ETH and 100 USDC, leaving us with 10 USDC, which we will leave outside the pool.

4.5 Two Intervals Around the Current Price

This strategy aims to fix the problem of having leftover liquidity. The first interval is the same as in the interval around the current price strategy 4.4, namely $[p_c - a, p_c + a]$. Let b be a paramter and $b \in [10, 1000]$, then the second interval will either be $[p_c, p_c + b]$ or $[p_c - b, p_c]$, depending on which asset is leftover [4]. This will allow us to use up all our liquidity.

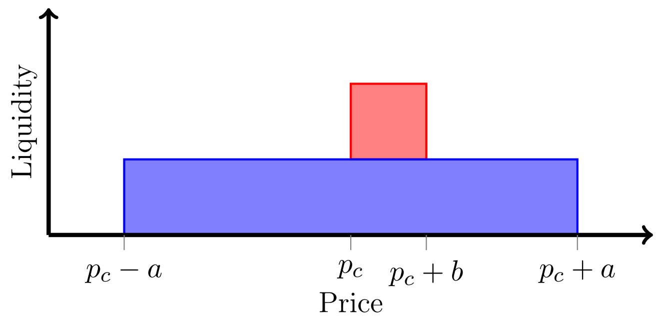

Figure 4.4: Two Intervals Around the Current Price Strategy

4.6 Fill Up

This is another strategy trying to solve the problem of 4.4. The first interval is $[p_c - a, p_c + a]$. The second interval is $[p_c, p_c + a]$ or $[p_c - a, p_c]$, depending on which asset is leftover.

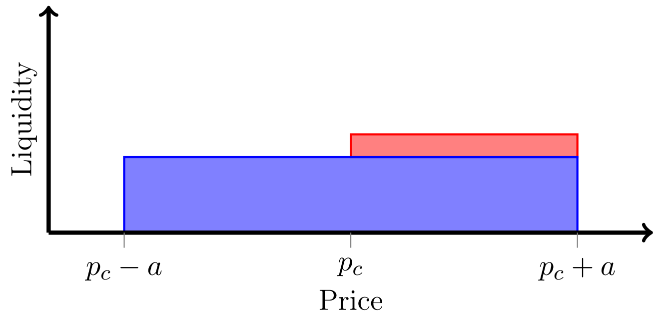

Figure 4.5: Fill Up Strategy

4.7 Swap

Another attempt to solve the problem of interval around the current price strategy 4.4 is to swap half of the excessive token for the other token. We would only swap half to ensure that the value we have for both tokens will be equal afterward. We provide liquidity in $[p_c - a, p_c + a]$. The disadvantage of this strategy is that we would have to pay fees to other LPs.

4.8 Range Order

This strategy is similar to the swap strategy 4.7, but we will use a range order as introduced in section 2.3.1 to swap half of the excessive token for the other token. Like the other strategies, we will rebalance every update interval. However, we will also do an additional rebalance after each update interval whenever the range order is executed.

The difference of this strategy compared to the two intervals around the current price strategy 4.5 with $b = 10$ is that this strategy only uses half of the remaining liquidity and does an additional rebalance.

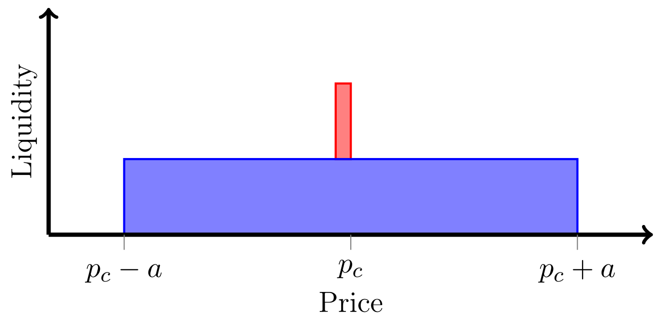

Figure 4.6: Range Order Strategy

4.9 Moving Average

This strategy does not look at the current price but instead at the moving average of the price p_a. We will consider different time intervals for calculating the moving average: two hours, six hours, one day, seven days, 30 days, 100 days, and 200 days. The liquidity position of this strategy will be $[p_a - a, p_a + a]$. As in the constant interval strategy 4.3, the leftover token will not be used to provide any liquidity.

4.10 Volatility sized Interval

In this strategy, we will adjust the size of the interval to the volatility. We consider different time intervals to calculate the past volatility. Those intervals are the same as in the moving average strategy 4.9 We will have one parameter $c \in (0, 64)$. To be more specific, we used a Q6.10 fixed point integer and tried every Q6.10 number from 1 to $2^{16} - 1$. Let v be the current standard deviation of the price, then the liquidity position will be $[p_c - v \cdot c, p_c + v \cdot c]$. The standard deviation was not scaled to the size of the history window, and it is the standard deviation of the price and not the price change. The leftover token will also not be used in this strategy.

4.11 Bollinger Bands

This strategy combines the moving average strategy 4.9 and the volatility sized intervals strategy 4.10. We will provide liquidity inside the Bollinger Bands. These bands are made up of a lower band $\text{BOLL} = p_a - c \cdot v$ and an upper band $\text{BOLU} = p_a + c \cdot v$. The liquidity position will be $[p_a - v \cdot c, p_a + v \cdot c]$. This strategy also has an unused leftover token. [23]

Chapter 5

Results

Snapshot

We will only present the results using daily snapshots (τ = one day from section 3.4.3). Our backtester also calculated volatility, VaR, and downward deviation for hourly and weekly snapshots.

Risk-Free Rate

The risk-free rate r_f will be used for the Sharpe ratio introduced in section 3.4.4 and Sortino Ratio introduced in section 3.4.6. As mentioned, we will use Aave's deposit rate. We got the data from [26] and took the average rate. The average rate was 2.62% for the in-sample period from May to November. For the out-of-sample period from November to February, the average rate was 3.81%.

5.1 Buying Ethereum

To have a comparison how the strategies performed, it makes sense to know how the price of Ethereum changed during the in-sample and out-of-sample period. If we bought Ethereum for 2$ at the beginning of the in-sample period and sold it at the end, we would have 3.175$. If we would have done the same in the out-of-sample period, we would have 1.216$.

5.2 Strategies

We will show the results per strategy. We will show the worst, median, and best parameters for strategies by annualized return, Sharpe ratio, and Sortino ratio. We will automatically compound the fees every update interval since a burn and the following collect function call automatically collects the fees.

The exception is the no provision strategy 5.2.1, which does not earn any fees. The v2 strategy 5.2.2 and constant interval strategy 5.2.3 always set the same interval. Thus, it makes sense to have non-compounded results for these two strategies. We will present for these two strategies compounded and non-compounded results

We will show the 1st, 50th, and 99th percentiles for all strategies that have parameters (strategies 5.2.3 to 5.2.11). The percentiles will be based on end amount and Sharpe ratio.

5.2.1 No Provision

The no provision strategy 4.1 gives the following results during the in-sample period.

	in	out
End Amount	$2.588	$1.608
Annulized Return	74.41%	-37.58%
Maximum Drawdown	-19.87%	-28.18%
Volatility	43.95%	41.60%
Downward Deviation	31.05%	32.98%
VaR 95%	-3.12%	-4.62%
Sharpe Ratio	1.633	-0.967
Sortino Ratio	2.312	-1.219

Table 5.1: No Provision

The bad performance in the out-of-sample period is due to the Ethereum price going down.

5.2.2 V2 Strategy

For the v2 strategy 4.2, we can either reinvest the fees each update interval or not reinvest any fees. We present the performance per update interval. ∞ means the update interval is infinity. In other words, we never reinvest our fees, and therefore it is non-compounding.

Update Interval	2 hours	6 hours	1 day	7 days	30 days	∞
End Amount	\$2.609	\$2.611	\$2.611	\$2.611	\$2.612	\$2.612
Annulized Return	77.59%	77.79%	77.85%	77.86%	77.99%	78.07%
Maximum Drawdown	-21.38%	-21.38%	-21.37%	-21.33%	-21.89%	-21.89%
Volatility	43.12%	43.13%	43.19%	43.21%	43.30%	43.16%
Downward Deviation	30.27%	30.28%	30.35%	30.43%	30.39%	30.54%
VaR 95%	-3.22%	-3.21%	-3.22%	-3.24%	-3.24%	-3.24%
Sharpe Ratio	1.739	1.743	1.742	1.741	1.741	1.748
Sortino Ratio	2.477	2.482	2.479	2.473	2.481	2.470

Table 5.2: V2 In-Sample

Update Interval	2 hours	6 hours	1 day	7 days	30 days	∞
End Amount	\$1.591	\$1.592	\$1.593	\$1.593	\$1.593	\$1.593
Annulized Return	-38.95%	-38.89%	-38.86%	-38.85%	-38.85%	-38.86%
Maximum Drawdown	-31.21%	-31.19%	-31.17%	-31.22%	-31.30%	-32.04%
Volatility	49.11%	49.15%	49.23%	49.30%	49.21%	49.20%
Downward Deviation	38.36%	38.40%	38.49%	38.64%	38.47%	38.70%
VaR 95%	-6.21%	-6.21%	-6.16%	-6.23%	-6.23%	-6.23%
Sharpe Ratio	-0.847	-0.844	-0.843	-0.841	-0.843	-0.843
Sortino Ratio	-1.084	-1.081	-1.078	-1.073	-1.078	-1.072

Table 5.3: V2 Out-Of-Sample

Compounding does not always give better results. While providing liquidity earns us fees, our capital is subject to impermanent loss. The difference between the update intervals is minimal. In the out-sample period, the strategy has a negative return due to Ethereum going down.

The V2 strategy is similar to the no provision strategy 5.2.1 because providing liquidity in such a large interval has minimal impermanent loss and earns minimal fees. In the out-of-sample period, the impermanent loss was greater than the earned fees.

5.2.3 Constant Interval

For the constant strategy, the interval is always the same. Thus, it is possible to have a non-compounding result.

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	∞	1 day	30 days	∞	1 day	30 days
Parameter a	320	23140	5680	320	23140	5860
End Amount	\$2.258	\$2.621	\$2.680	\$1.352	\$1.585	\$1.537
Annulized Return	29.90%	79.35%	88.20%	-57.04%	-39.45%	-43.40%
Maximum Drawdown	-36.63%	-22.07%	-28.06%	-48.75%	-32.56%	-40.36%
Volatility	58.35%	43.05%	46.26%	91.04%	53.06%	76.53%
Downward Deviation	42.35%	30.17%	31.78%	70.97%	41.26%	58.39%
VaR 95%	-5.44%	-3.36%	-3.65%	-10.46%	-6.59%	-8.16%
Sharpe Ratio	0.468	1.782	1.850	-0.655	-0.793	-0.601
Sortino Ratio	0.644	2.543	2.693	-0.841	-1.020	-0.788

Table 5.4: Constant Interval by End Amount

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	∞	2 hours	∞	∞	2 hours	∞
Parameter a	270	23020	7710	270	23020	7710
End Amount	\$2.259	\$2.619	\$2.666	\$1.353	\$1.584	\$1.560
Annulized Return	30.14%	79.02%	85.98%	-57.00%	-39.57%	-41.52%
Maximum Drawdown	-36.63%	-22.08%	-26.20%	-48.65%	-32.62%	-40.28%
Volatility	58.46%	42.94%	43.95%	90.92%	52.94%	69.05%
Downward Deviation	42.42%	30.05%	31.18%	70.84%	41.11%	53.59%
VaR 95%	-5.44%	-3.36%	-3.64%	-10.43%	-6.52%	-7.67%
Sharpe Ratio	0.471	1.779	1.897	-0.656	-0.797	-0.639
Sortino Ratio	0.649	2.543	2.673	-0.842	-1.026	-0.824

Table 5.5: Constant Interval by Sharpe Ratio

This strategy's median and 99th percentile had slightly higher returns than the no provision strategy 5.2.1 during the in-sample period. However, the contrary happened when the Ethereum price declined during the out-of-sample period. The performance of this strategy is slightly worse than the no provision strategy during the out-of-sample period. The first percentile parameter has a small interval and performs poorly in the in- and out-of-sample period.

If we look at the Sharpe ratio, we get a similar picture. We can see that this strategy's median and 99th percentile performed slightly better than the no provision strategy 5.2.1 during the in-sample period. However, the return is negative during the out-of-sample period, and it does not make sense to compare Sharpe ratios.

5.2.4 Interval Around the Current Price

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	2 hours	30 days	6 hours	2 hours	30 days	6 hours
Parameter a	38360	15900	1950	38360	15900	1950
End Amount	\$2.614	\$2.628	\$3.457	\$1.588	\$1.579	\$1.354
Annulized Return	78.33%	80.38%	225.99%	-39.24%	-40.00%	-56.95%
Maximum Drawdown	-21.64%	-23.56%	-31.52%	-31.70%	-33.53%	-46.84%
Volatility	43.14%	43.43%	78.83%	50.41%	55.26%	88.07%
Downward Deviation	30.25%	30.32%	54.66%	39.30%	42.86%	68.19%
VaR 95%	-3.27%	-3.53%	-6.14%	-6.36%	-6.81%	-10.28%
Sharpe Ratio	1.755	1.791	2.833	-0.830	-0.771	-0.676
Sortino Ratio	2.503	2.565	4.087	-1.065	-0.994	-0.874

Table 5.6: Interval Around the Current Price by End Amount

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	30 days	2 hours	7 days	30 days	2 hours	7 days
Parameter a	38870	21440	2590	38870	21440	3590
End Amount	\$2.616	\$2.627	\$3.415	\$1.590	\$1.581	\$1.477
Annulized Return	78.53%	80.25%	217.53%	-39.08%	-39.85%	-48.06%
Maximum Drawdown	-22.23%	-22.15%	-29.20%	-31.78%	-32.66%	-41.13%
Volatility	43.28%	43.31%	73.39%	50.47%	53.01%	80.78%
Downward Deviation	30.34%	30.30%	50.62%	39.38%	41.18%	61.97%
VaR 95%	-3.30%	-3.39%	-5.77%	-6.41%	-6.56%	-8.84%
Sharpe Ratio	1.754	1.792	2.928	-0.826	-0.801	-0.627
Sortino Ratio	2.502	2.562	4.246	-1.059	-1.031	-0.818

Table 5.7: Interval Around the Current Price by Sharpe Ratio

Choosing a relatively small interval (1950 ticks = 19.5%) gives us an excellent performance during the in-sample period. However, during the out-of-sample period, the return is worse than the reference strategy (no provision strategy 5.2.1), but the difference is not as big as in the in-sample period.

The strategies that perform well regarding the end amount also perform well regarding the Sharpe ratio. Furthermore, smaller intervals have more volatility than bigger intervals.

5.2.5 Two Intervals the Around Current Price

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	7 days	2 hours	30 days	7 days	2 hours	30 days
Parameter a	850	11130	13880	850	11130	13880
Parameter b	440	850	900	440	850	900
End Amount	\$1.676	\$2.577	\$2.649	\$1.834	\$1.577	\$1.585
Annulized Return	-31.73%	72.85%	83.46%	-17.03%	-40.12%	-39.50%
Maximum Drawdown	-36.66%	-21.56%	-23.91%	-30.87%	-31.46%	-33.82%
Volatility	58.31%	42.96%	46.94%	65.77%	49.60%	56.34%
Downward Deviation	47.04%	30.54%	33.07%	51.04%	39.12%	43.63%
VaR 95%	-5.66%	-3.20%	-3.59%	-6.15%	-6.37%	-6.91%
Sharpe Ratio	-0.589	1.635	1.722	-0.299	-0.862	-0.748
Sortino Ratio	-0.730	2.300	2.444	-0.385	-1.093	-0.966

Table 5.8: Two Intervals Around the Current Price by End Amount

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	7 days	1 day	30 days	7 days	1 day	30 days
Parameter a	790	34950	32690	790	34950	32690
Parameter b	590	190	920	590	190	920
End Amount	\$1.665	\$2.566	\$2.622	\$1.641	\$1.548	\$1.591
Annulized Return	-32.68%	71.23%	79.54%	-34.80%	-42.48%	-39.00%
Maximum Drawdown	-35.66%	-21.44%	-22.38%	-39.07%	-31.46%	-31.95%
Volatility	57.41%	42.71%	44.03%	73.89%	49.24%	50.99%
Downward Deviation	46.34%	30.44%	30.95%	59.25%	39.28%	39.75%
VaR 95%	-5.66%	-3.37%	-3.30%	-9.77%	-6.29%	-6.46%
Sharpe Ratio	-0.615	1.607	1.747	-0.507	-0.916	-0.816
Sortino Ratio	-0.762	2.254	2.486	-0.632	-1.148	-1.047

Table 5.9: Two Intervals Around the Current Price by Sharpe Ratio

The return of this strategy is similar to that of the no provision strategy 5.2.1 for the 50th and 99th percentiles. However, choosing bad parameters can lead to significantly worse performance.

5.2.6 Fill Up

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	6 hours	6 hours	30 days	6 hours	6 hours	30 days
Parameter a	450	26890	4670	450	26890	4670
End Amount	\$1.691	\$2.600	\$2.741	\$1.099	\$1.590	\$1.529
Annulized Return	-30.36%	76.25%	97.51%	-72.56%	-39.06%	-44.02%
Maximum Drawdown	-37.95%	-21.70%	-29.59%	-48.38%	-31.73%	-39.19%
Volatility	50.53%	42.81%	49.13%	65.97%	50.23%	72.94%
Downward Deviation	44.02%	30.10%	33.31%	61.10%	39.25%	55.79%
VaR 95%	-4.80%	-3.22%	-3.85%	-10.23%	-6.40%	-8.23%
Sharpe Ratio	-0.653	1.720	1.931	-1.140	-0.830	-0.639
Sortino Ratio	-0.749	2.446	2.849	-1.230	-1.062	-0.836

Table 5.10: Fill Up by End Amount

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	7 days	2 hours	30 days	7 days	2 hours	30 days
Parameter a	250	35650	4210	250	35650	4210
End Amount	\$1.674	\$2.601	\$2.756	\$2.306	\$1.590	\$1.521
Annulized Return	-31.91%	76.39%	99.82%	35.95%	-39.07%	-44.64%
Maximum Drawdown	-37.84%	-21.54%	-30.47%	-22.01%	-31.46%	-39.78%
Volatility	60.22%	42.95%	51.02%	69.47%	49.57%	75.05%
Downward Deviation	48.68%	30.18%	34.44%	48.44%	38.74%	57.33
VaR 95%	-5.68%	-3.20%	-3.98%	-5.69%	-6.32%	-8.50%
Sharpe Ratio	-0.573	1.718	1.905	0.480	-0.841	-0.630
Sortino Ratio	-0.709	2.445	2.822	0.688	-1.076	-0.824

Table 5.11: Fill Up by Sharpe Ratio

This strategy performed similarly to the two intervals around price strategy 5.2.5 for the 50th and 99th percentile. Choosing the wrong parameter (a = 450) can lead to terrible performance, both in the in-sample and the out-of-sample period. Interestingly, for a = 250, the return during the out-of-sample period is positive.

5.2.7 Swap

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	7 days	1 day	30 days	7 days	1 day	30 days
Parameter a	250	23480	8000	250	23480	8000
End Amount	\$1.866	\$2.601	\$2.615	\$1.637	\$1.584	\$1.633
Annulized Return	-13.87%	76.45%	78.47%	-35.11%	-39.60%	-35.50%
Maximum Drawdown	-33.94%	-21.40%	-26.00%	-41.80%	-31.30%	-32.80%
Volatility	61.05%	43.26%	48.22%	75.91%	49.42%	54.99%
Downward Deviation	48.96%	30.59%	34.11%	59.46%	38.86%	42.34%
VaR 95%	-5.66%	-3.23%	-3.91%	-10.32%	-6.21%	-6.74%
Sharpe Ratio	-0.270	1.707	1.573	-0.497	-0.854	-0.693
Sortino Ratio	-0.337	2.413	2.223	-0.635	-1.087	-0.900

Table 5.12: Swap by End Amount

This strategy performed similarly to the two intervals around price strategy 5.2.5 for the 50th and 99th percentile. Choosing small intervals can even lead to negative returns in the in-sample period. The negative returns are probably due to the strategy swapping a lot and paying fees each time. The strategy needs to swap more if the interval width is small. The narrower the liquidity position, the more the ratio of your assets change as the price changes.

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	7 days	6 hours	7 days	7 days	6 hours	7 days
Parameter a	500	19570	38060	500	19570	38060
End Amount	\$1.861	\$2.598	\$2.609	\$1.637	\$1.585	\$1.593
Annulized Return	-14.43%	75.89%	77.63%	-35.14%	-39.49%	-38.85%
Maximum Drawdown	-33.59%	-21.49%	-21.34%	-40.49%	-31.24%	-31.26%
Volatility	58.47%	43.11%	43.25%	71.89%	49.27%	49.44%
Downward Deviation	47.23%	30.35%	30.52%	57.36%	38.61%	38.80%
VaR 95%	-5.66%	-3.22%	-3.21%	-9.92%	-6.28%	-6.26%
Sharpe Ratio	-0.292	1.699	1.734	-0.525	-0.855	-0.839
Sortino Ratio	-0.361	2.414	2.457	-0.658	-1.091	-1.069

Table 5.13: Swap by Sharpe Ratio

5.2.8 Range Order

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	7 days	2 hours	30 days	7 days	2 hours	30 days
Parameter a	50	23780	25590	50	23780	25590
End Amount	\$1.837	\$2.589	\$2.612	\$1.605	\$1.583	\$1.593
Annulized Return	-16.75%	74.69%	78.01%	-37.83%	-39.62%	-38.80%
Maximum Drawdown	-28.67%	-21.49%	-22.67%	-42.69%	-31.27%	-32.32%
Volatility	45.48%	43.07%	43.77%	79.20%	49.23%	51.97%
Downward Deviation	36.06%	30.30%	30.78%	63.29%	38.56%	40.44%
VaR 95%	-4.06%	-3.22%	-3.34%	-10.55%	-6.26%	-6.55%
Sharpe Ratio	-0.426	1.673	1.722	-0.511	-0.858	-0.797
Sortino Ratio	-0.537	2.379	2.449	-0.639	-1.095	-1.024

Table 5.14: Range Order by End Amount

	1%	50%	99%			
Sample	in	in	in	out	out	out
Update Interval	1 day	7 days	30 days	1 day	7 days	30 days
Parameter a	940	24570	38030	940	24570	38030
End Amount	\$1.835	\$2.585	\$2.612	\$1.222	\$1.602	\$1.593
Annulized Return	-16.93%	74.10%	77.98%	-65.49%	-38.04%	-38.83%
Maximum Drawdown	-28.12%	-21.45%	-22.25%	-44.38%	-31.44%	-31.78%
Volatility	45.09%	42.82%	43.49%	62.02%	49.96%	50.48%
Downward Deviation	38.91%	30.42%	30.55%	55.21%	39.20%	39.37%
VaR 95%	-3.79%	-3.21%	-3.30%	-8.03%	-6.47%	-6.42%
Sharpe Ratio	-0.434	1.669	1.733	-1.098	-0.814	-0.821
Sortino Ratio	-0.502	2.350	2.466	-1.234	-1.037	-1.053

Table 5.15: Range Order by Sharpe Ratio

Even though this strategy does not have to pay exchange fees, the performance is similar to the swap strategy 5.2.7. Small intervals did not perform well in both the in- and out-of-sample period. We could not determine why.

5.2.9 Moving Average

	1%	50%	99%			
Update Interval	1 day	2 hours	2 hours	1 day	2 hours	2 hours
History Window	200 days	30 days	2 hours	200 days	30 days	2 hours
Parameter a	3170	24160	2180	3170	24160	2180
End Amount	$2.314	$2.625	$3.447	$1.579	$1.584	$1.369
Annulized Return	36.99%	79.90%	224.12%	-39.99%	-39.56%	-55.92%
Maximum Drawdown	-19.85%	-22.01%	-30.94%	-39.26%	-32.40%	-46.06%
Volatility	29.18%	43.28%	76.77%	72.16%	52.36%	86.65%
Downward Deviation	20.64%	30.29%	53.03%	55.17%	40.70%	67.03%
VaR 95%	-2.61%	-3.35%	-6.02%	-7.59%	-6.49%	-10.10%
Sharpe Ratio	1.178	1.786	2.885	-0.591	-0.806	-0.676
Sortino Ratio	1.665	2.551	4.177	-0.772	-1.037	-0.873

Table 5.16: Moving Average by End Amount

	1%	50%	99%			
Update Interval	6 hours	7 days	2 hours	6 hours	7 days	2 hours
History Window	100 days	7 days	6 hours	100 days	7 days	6 hour
Parameter a	1540	24950	2350	1540	24950	2350
End Amount	$2.343	$2.628	$3.429	$1.386	$1.586	$1.382
Annulized Return	40.73%	80.29%	220.44%	-54.72%	-39.42%	-55.02%
Maximum Drawdown	-32.70%	-21.89%	-30.47%	-45.53%	-32.32%	-45.45%
Volatility	56.21%	43.45%	74.55%	86.48%	52.38%	85.59%
Downward Deviation	41.01%	30.54%	51.36%	67.09%	40.90%	66.14%
VaR 95%	-5.02%	-3.37%	-5.76%	-10.61%	-6.56%	-9.95%
Sharpe Ratio	0.678	1.788	2.922	-0.663	-0.803	-0.673
Sortino Ratio	0.929	2.543	4.241	-0.855	-1.028	-0.872

Table 5.17: Moving Average by Sharpe Ratio

This strategy performs very similarly to the interval around price strategy 5.2.4. The strategy with the smallest history windows performed the best in the in-sample period. In contrast, the strategies with the largest history window performed the best in the out-of-sample period. A strategy like the 99th percentile strategy according to the end amount, with a history window of 2 hours, is almost the same strategy as the interval around price strategy 5.2.4, because the moving average of the last 2 hours is almost the same as the current price.

5.2.10 Volatility Sized Interval

	1%	50%	99%			
Update Interval	30 days	6 hours	1 day	30 days	6 hours	1 day
History Window	1 day	1 day	7 days	1 day	1 day	7 days
MultiplierX10	19962	18354	6364	19962	18354	6364
Parameter c	19.494	17.924	6.215	19.494	17.924	6.215
End Amount	\$2.353	\$2.646	\$3.439	\$1.483	\$1.379	\$1.552
Annulized Return	42.12%	83.06%	222.47%	-47.59%	-55.17%	-42.20%
Maximum Drawdown	-26.51%	-29.56%	-31.28%	-41.73%	-45.48%	-35.72%
Volatility	39.78%	51.68%	80.99%	81.26%	83.60%	62.80%
Downward Deviation	27.32%	36.14%	56.47%	61.77%	64.80%	48.42%
VaR 95%	-3.24%	-4.33%	-6.14%	-9.15%	-9.70%	-7.15%
Sharpe Ratio	0.993	1.556	2.715	-0.618	-0.691	-0.714
Sortino Ratio	1.446	2.226	3.893	-0.813	-0.892	-0.926

Table 5.18: Volatility Sized Interval by end amount

	1%	50%	99%			
Update Interval	7 days	7 days	1 day	7 days	7 days	1 day
History Window	1 day	100 days	2 hours	1 day	100 days	2 hours
MultiplierX10	15460	27317	11977	15460	27317	11977
Parameter c	15.098	26.677	11.696	15.098	26.677	11.696
End Amount	$2.380	$2.621	$3.235	$1.405	$1.593	$1.479
Annulized Return	45.61%	79.40%	182.57%	-53.37%	-38.85%	-47.88%
Maximum Drawdown	-25.89%	-21.48%	-29.53%	-44.15%	-31.22%	-40.98%
Volatility	42.98%	43.71%	61.79%	86.27%	49.30%	80.52%
Downward Deviation	30.30%	30.77%	42.09%	66.58%	38.64%	61.84%
VaR 95%	-3.32%	-3.29%	-5.07%	-9.99%	-6.23%	-9.48%
Sharpe Ratio	1.000	1.757	2.912	-0.649	-0.841	-0.627
Sortino Ratio	1.419	2.495	4.275	-0.841	-1.073	-0.817

Table 5.19: Volatility Sized Interval by Sharpe Ratio

The performance of this strategy is also similar to the interval around price strategy 5.2.4. The smallest history windows performed the best during the in-sample period. A small history window leads to a smaller standard deviation and, therefore, a smaller interval. We did not scale the standard deviation with the history window.

5.2.11 Bollinger Bands

	1%	50%	99%			
Update Interval	30 days	1 day	1 day	30 days	1 day	1 day
History Window	6 hours	2 hours	6 hours	6 hours	2 hours	6 hours
MultiplierX10	15474	35762	4356	15474	35762	4356
Parameter c	15.111	34.924	4.254	15.111	34.924	4.254
End Amount	\$2.354	\$2.640	\$3.442	\$1.461	\$1.515	\$1.541
Annulized Return	42.24%	82.15%	223.06%	-49.23%	-45.13%	-43.05%
Maximum Drawdown	-27.45%	-26.66%	-32.24%	-42.27%	-38.79%	-38.77%
Volatility	41.33%	47.13%	83.19%	83.76%	72.90%	81.54%
Downward Deviation	28.41%	32.82%	58.14%	63.61%	55.86%	62.43%
VaR 95%	-3.32%	-3.84%	-6.43%	-9.49%	-8.29%	-8.52%
Sharpe Ratio	0.959	1.687	2.650	-0.619	-0.655	-0.560
Sortino Ratio	1.394	2.424	3.791	-0.815	-0.855	-0.732

Table 5.20: Bollinger Bands by End Amount

	1%	50%	99%			
Update Interval	30 days	7 days	7 days	30 days	7 days	7 days
History Window	2 hours	200 days	7 days	2 hours	200 days	7 days
MultiplierX10	13918	5427	9610	13918	5427	9610
Parameter c	13.592	5.300	9.385	13.592	5.300	9.385
End Amount	\$2.372	\$2.617	\$3.373	\$1.453	\$1.593	\$1.571
Annulized Return	44.51%	78.71%	209.25%	-49.84%	-38.84%	-40.65%
Maximum Drawdown	-27.86%	-21.43%	-27.01%	-42.47%	-31.22%	-34.36%
Volatility	42.14%	43.50%	70.83%	84.73%	49.32%	58.43%
Downward Deviation	28.97%	30.63%	48.87%	64.31%	38.65%	45.31%
VaR 95%	-3.36%	-3.27%	-5.47%	-9.63%	-6.23%	-6.94%
Sharpe Ratio	0.994	1.749	2.917	-0.619	-0.841	-0.741
Sortino Ratio	1.446	2.484	4.228	-0.816	-1.073	-0.955

Table 5.21: Bollinger Bands by Sharpe Ratio

This strategy performs similarly to the interval around price strategy 5.2.4 during the in-sample period. However, the return during the out-of-sample period is better. It is almost as good in the out-of-sample period as the no provision strategy 5.2.1. This strategy worked pretty well overall.

5.3 Detailed Analysis of the Interval Around the Current Price Strategy

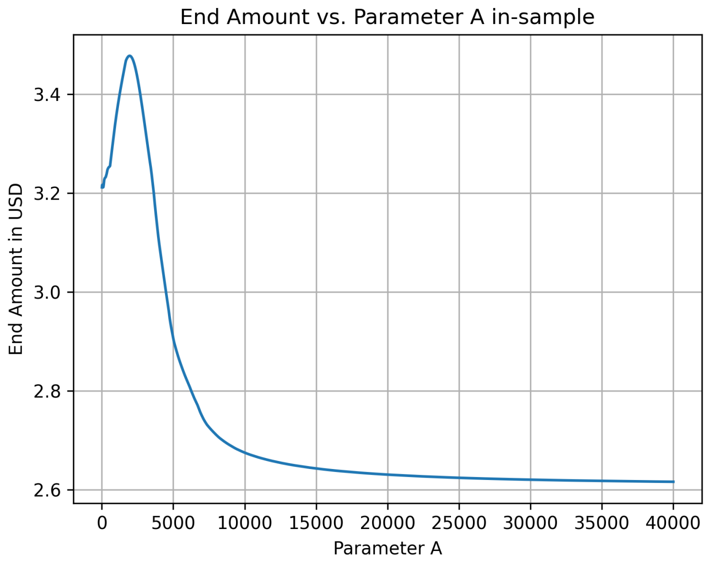

Figure 5.1: Interval Around the Current Price Strategy In-Sample

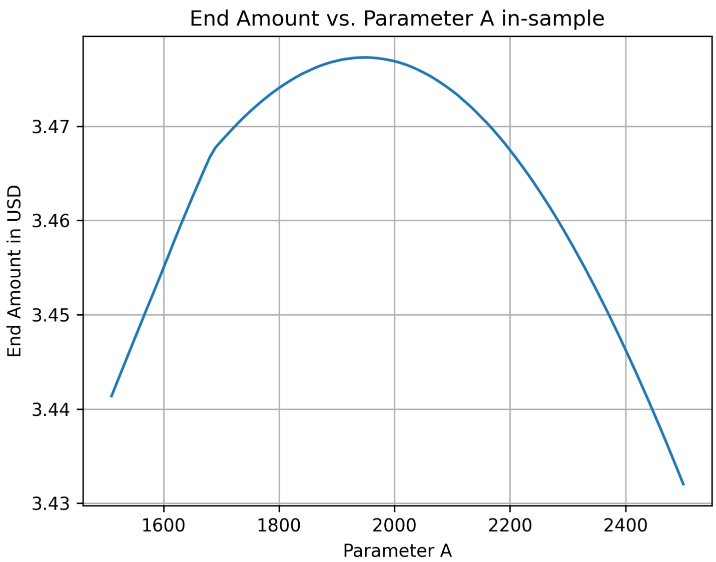

Figure 5.2: Detailed View of Figure 5.1

As we can see in Figure 5.1 and Figure 5.2 the strategy has a single local maximum at a = 1950 for the in-sample period. The difference between the bigger intervals is minimal.

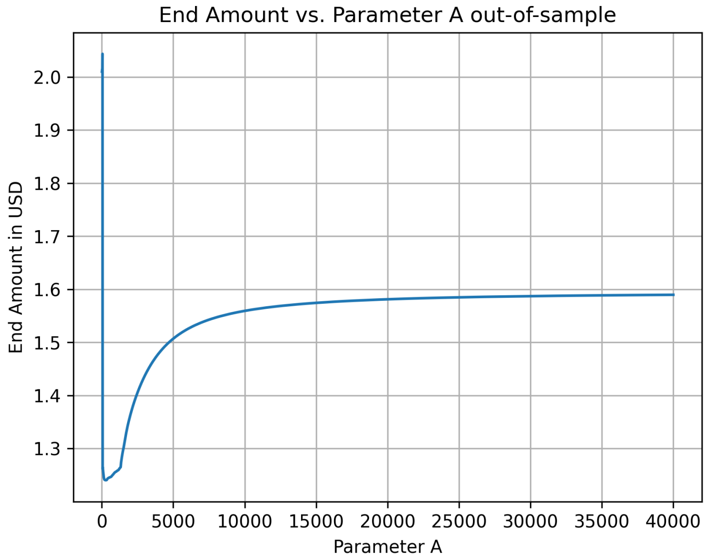

Figure 5.3: IInterval Around the Current Price Strategy Out-Of-Sample

In the out-of-sample period, there is a single local minimum at a = 280. The curve of the out-of-sample period seems mirrored to that of the in-sample period. Small intervals tend to perform poorly. The tiny intervals ($a < 50$) have a decent return because all their liquidity will be exchanged to USDC initially. Since there is no ETH left, it is impossible to provide liquidity anymore.

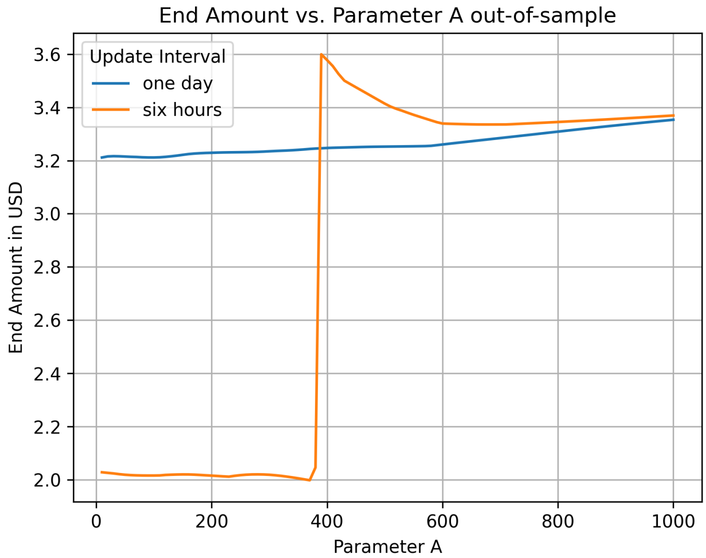

Figure 5.4: End amount with Different Time Intervals In-Sample

The Figure 5.4 shows the difference between returns depending on the update intervals. We can see a sudden jump at 380 in the orange graph. The reason for the jump is that for all parameters $a \leq 380$, all liquidity will be in USDC, and for $a \geq 390$, the liquidity will be in Ethereum. The price of Ethereum is higher at the end of the in-sample period than at the beginning, which leads to good performance, because it is as if we had bought Ethereum from the beginning until the lowest price point of Ethereum. Once we only have Ethereum left, all our assets will stay in Ethereum, because we cannot provide any liquidity since no USDC is left. In the interval around price strategy 4.4, we have to provide a symmetric interval around the current price, so we need an equal value of both tokens. We can not provide any liquidity, if one the amount of one asset is zero. Thus, our strategy is often very similar to either buying or selling Ethereum. All positions that need both assets to provide liquidity suffer from this problem. Most of our strategies only have a single position that would need both assets.

5.3.1 Monthly Analysis

We will show the end amount for all parameters A for the months June 2021 until February 2022. At each month, we start with 1USDC and 1$ worth of ETH.

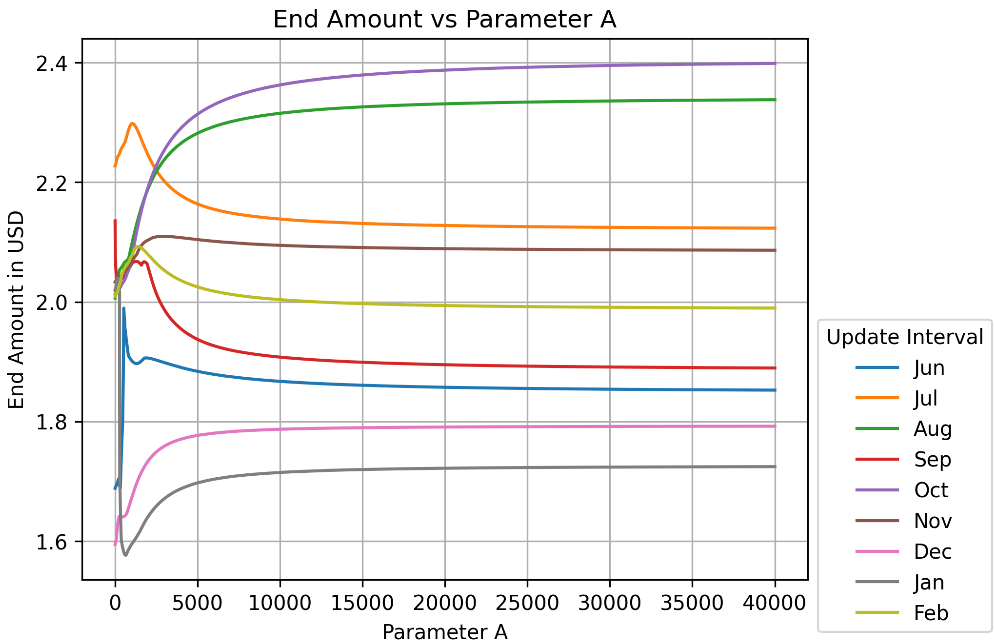

Figure 5.5: Interval Around the Current Price Strategy Monthly

What is surprising is that some curves are concav and some are convex. We can not predict the curvature based on the end amount.

5.4 Detailed Analysis of the Fill up Strategy

The fill up strategy can always provide liquidity, because one of the intervals only needs one asset. We will show the results for an update interval of one day and 30 days.

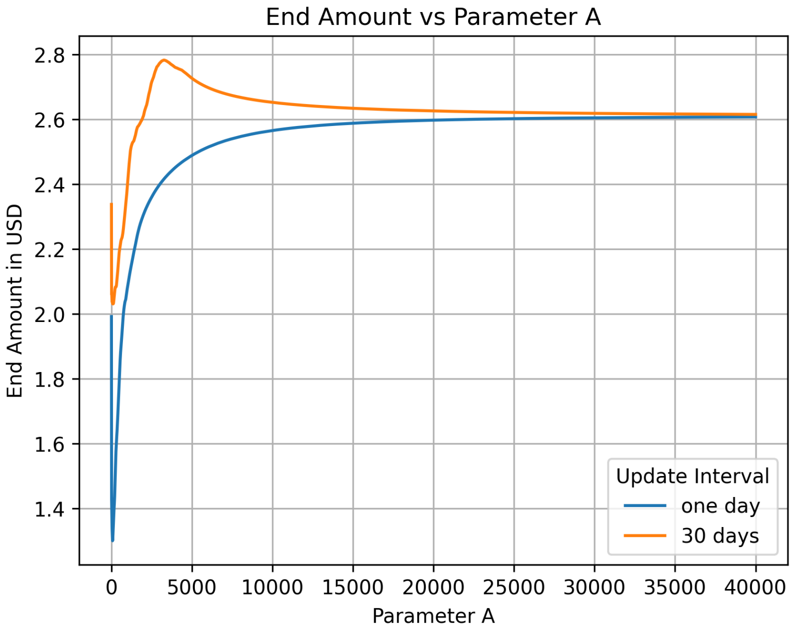

Figure 5.6: Fill Up Strategy In-Sample

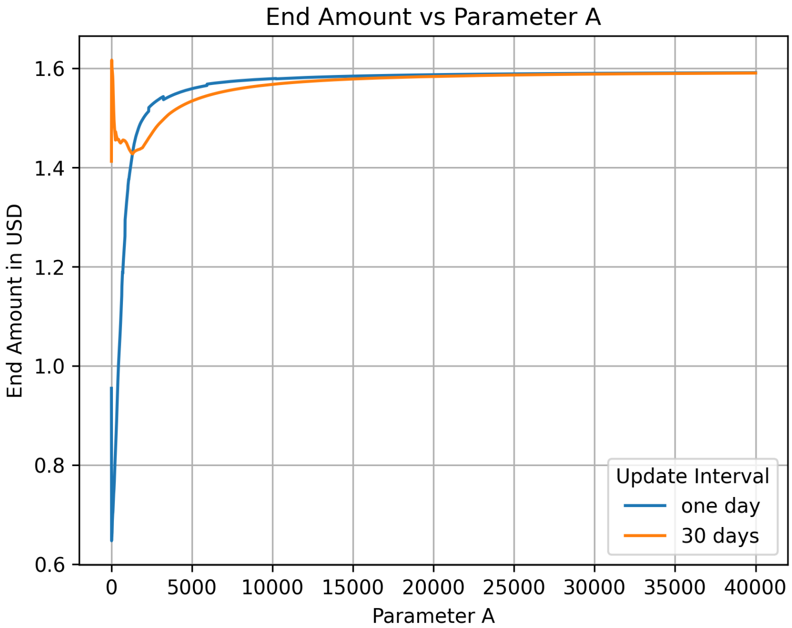

Figure 5.7: Fill Up Strategy Out-Of-Sample

We can see, this strategy does not work for an update interval of one day, as the largest intervals produce the best results, and those results are close to the v2 strategy. With an update interval of 30 days, the strategy worked and we got a better result than the v2 strategy. The best parameter here was 3270 and with that the strategy got an end amount of 2.783$.

5.5 Delta

In this section, we will show the delta of various strategies. We will use an update interval of one day for all strategies.

5.5.1 Constant Interval

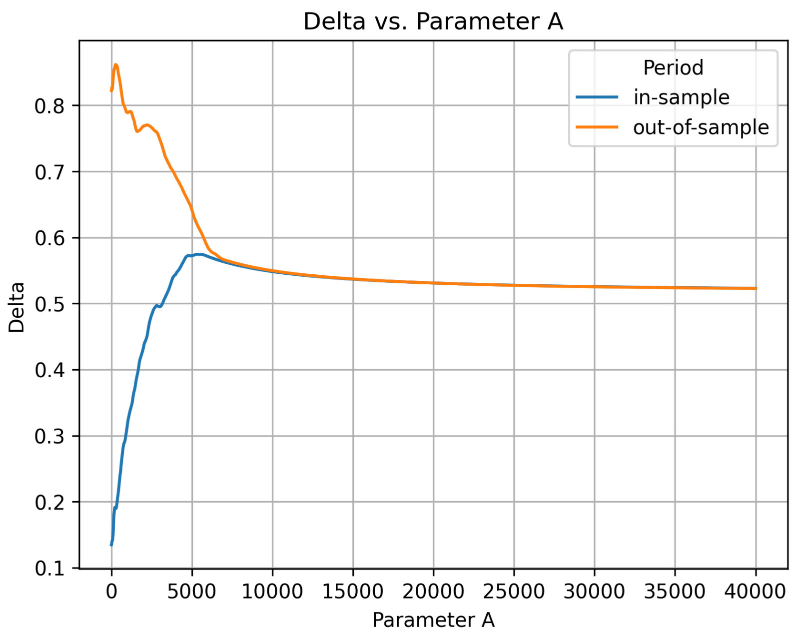

Figure 5.8: Delta of Constant Interval Strategy

5.5.2 Interval Around the Current Price

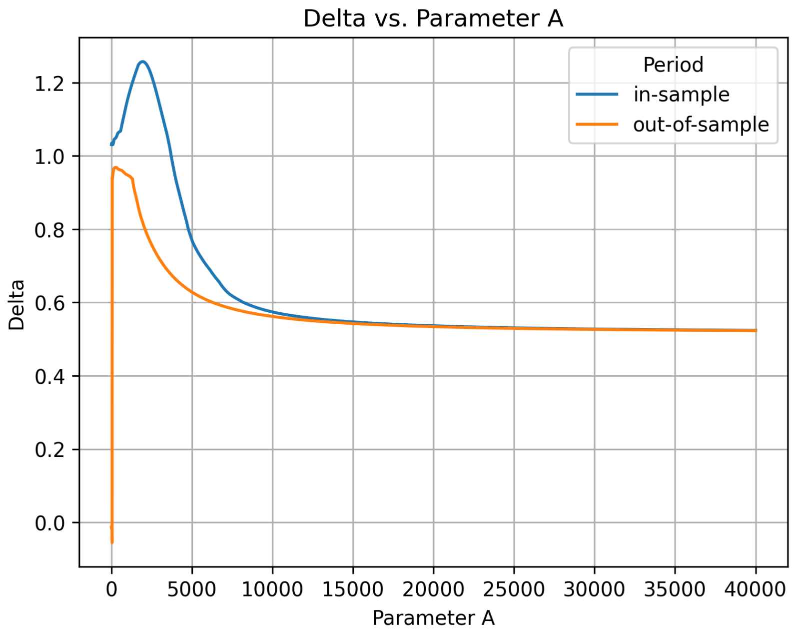

Figure 5.9: Delta of Interval Around the Current Price Strategy

5.5.3 Range Order

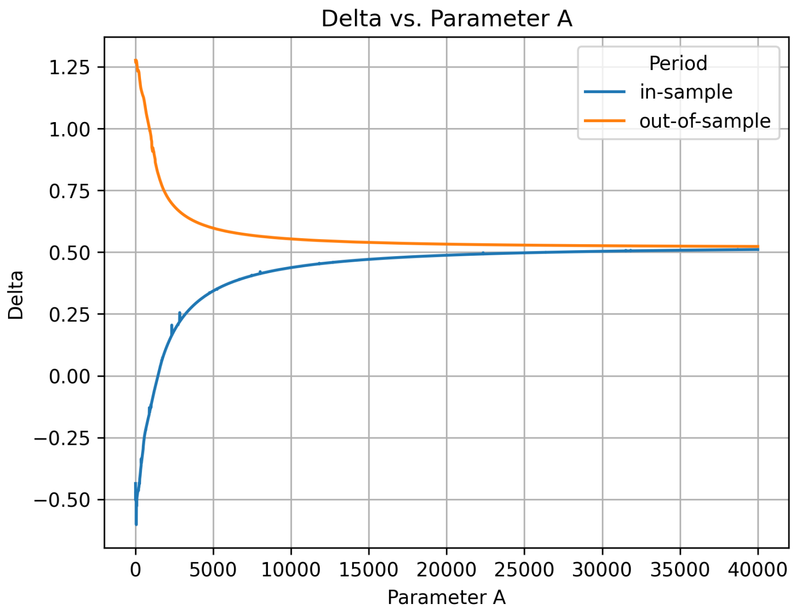

Figure 5.10: Delta of Range Order Strategy

5.5.4 Bollinger Bands

We will use a history interval of one day for this strategy.

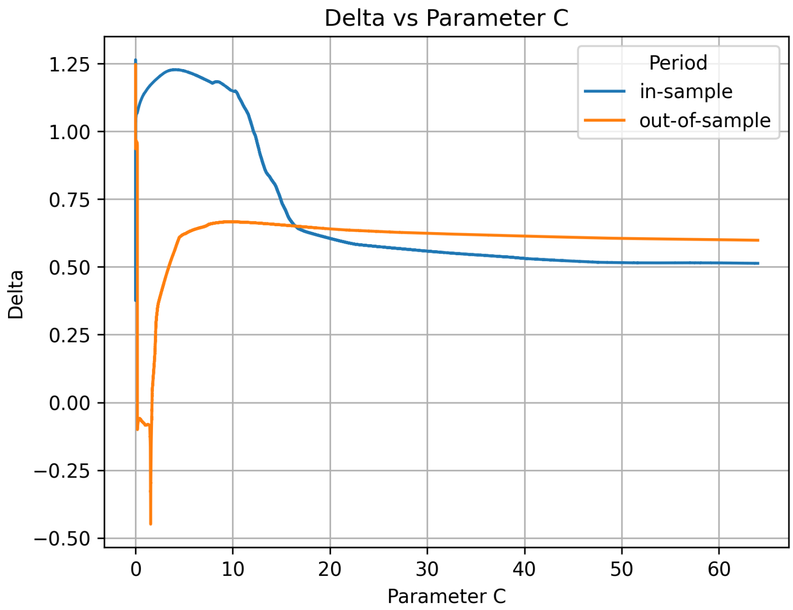

Figure 5.11: Delta of Bollinger Bands Strategy

5.5.5 Comparisons Between Strategies

In the Figures 5.8 to 5.11 we see that all strategies have a delta of around 0.5 for large intervals, because proving liquidity in a large interval is similar to just holding 50% of your portfolio in Ethereum and never providing any liquidity. The delta of doing that is 0.5. For almost all parameters of the four analyzed strategies, the delta was greater than zero. The interval around the current price strategy and Bollinger Bands strategy worked pretty well. They have a higher delta when the price is going up.

Chapter 6

Conclusion and Future Work

We have built a robust backtesting tool for Uniswap v3 that can be reused. With that tool, we analyzed eleven different strategies and have found strategies that performed well when the price of Ethereum was rising. However, often these strategies performed poorly when the price of Ethereum sunk. We have found strategies that have a bigger delta when the price was rising than when it was falling. These are strategies that could be implemented. In future work, other stablecoin-Ethereum pools should also be analyzed. However, we think that the results will still heavily depend on the Ethereum price. Furthermore, it would also be interesting to analyze pools like BTC-ETH.

We identified two categories of strategies, one that perform similarly to the v2 strategy, namely no provision, constant interval, two intervals around the current price, fill up, swap, and range order strategy. The other category of strategies performed similar to the interval around the current price strategy. Those strategies are moving average, volatility sized interval, and Bollinger Bands strategy. In future work, it would be interesting to combine the strategies, e.g. Bollinger Bands with fill up or swap.

Future work should instead analyze stablecoin pairs or find a way to hedge using different DeFi protocols and create a delta-neutral strategy. Furthermore, it would also be interesting to compare liquidity provisions on Uniswap v3 with different DEXs. With Uniswap launching on Arbitrum, Optimism, and Polygon, it would be compelling to compare pools on these platforms to pools on the Ethereum. However, these platforms do not have alot of volume yet.

Bibliography

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., Oct. 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] V. Buterin, "Ethereum whitepaper," Tech. Rep., 2013. [Online]. Available: https://ethereum.org/en/whitepaper/

[3] H. Adams, N. Zinsmeister, M. Salem, R. Keefer, and D. Robinson, "Uniswap v3 whitepaper," Tech. Rep., Mar. 2021. [Online]. Available: https://uniswap.org/whitepaper-v3.pdf

[4] Max, "Introducing alpha vaults — an lp strategy for uniswap v3," May 2021. [Online]. Available: https://medium.com/charmfinance/introducing-alpha-vaults-an-lp-strategy-for-uniswap-v3-ebf500b67796

[5] G. Strategies, "Gamma's strategy one," 2021. [Online]. Available: https://github.com/GammaStrategies/strategy-one

[6] J. Clark, "The replicating portfolio of a constant product market," Apr. 2020. [Online]. Available: https://ssrn.com/abstract=3550601

[7] J. Clark, "The replicating portfolio of a constant product market with bounded liquidity," Aug. 2021. [Online]. Available: https://ssrn.com/abstract=3898384

[8] *Strategic Liquidity Provision in Uniswap v3*, june 2021. [Online]. Available: https://arxiv.org/abs/2106.12033

[9] R. Fritsch, "Concentrated liquidity in automated market makers," in *Proceedings of the 2021 ACM CCS Workshop on Decentralized Finance and Security*, ser. DeFi '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 15–20. [Online]. Available: https://doi.org/10.1145/3464967.3488590

[10] M. team, "Programming amms on top of univ3," Apr. 2021. [Online]. Available: https://mellow.finance/research.pdf

[11] "Erc-20 token standard," Tech. Rep. [Online]. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

[12] "Uniswap v2 analytics." [Online]. Available: https://v2.info.uniswap.org/home

[13] "Uniswap v3 analytics." [Online]. Available: https://info.uniswap.org/home

[14] H. Adams, N. Zinsmeister, and D. Robinson, "Uniswap v2 whitepaper," Tech. Rep., Mar. 2020. [Online]. Available: https://uniswap.org/whitepaper.pdf

[15] *Formal Specification of Constant Product* $x \times y = k$, Oct. 2018. [Online]. Available: https://github.com/runtimeverification/verified-smart-contracts/blob/uniswap/uniswap/x-y-k.pdf

[16] "Uniswap v3 subgraph." [Online]. Available: https://thegraph.com/hosted-service/subgraph/uniswap/uniswap-v3

[17] "Etherscan api." [Online]. Available: https://etherscan.io/apis

[18] "Loanscan." [Online]. Available: https://www.centre.io/usdc-transparency

[19] "Hardhat." [Online]. Available: https://hardhat.org/

[20] "Uniswap v3 core smart contract." [Online]. Available: https://github.com/Uniswap/v3-core

[21] "Uniswap v3 sdk." [Online]. Available: https://github.com/Uniswap/v3-sdk

[22] Daoleno, "Uniswap v3 sdk go." [Online]. Available: https://github.com/daoleno/uniswapv3-sdk

[23] "Investopedia." [Online]. Available: https://www.investopedia.com/

[24] J. C. Hull, *Options, Futures and other Derivatives.* Pearson, 2015.

[25] T. N. Rollinger and S. T. Hoffman, "Sortino: A 'sharper' ratio." [Online]. Available: https://www.cmegroup.com/education/files/rr-sortino-a-sharper-ratio.pdf

[26] "Loanscan." [Online]. Available: https://loanscan.io/earn/historical