Automated Market Makers: A Stochastic Optimization Approach for Profitable Liquidity Concentration

Simon Caspar Zeller*, Paul-Niklas Ken Kandora*, Daniel Kirste†, Niclas Kannengießer*
Steffen Rebennack*, Ali Sunyaev†

*Karlsruhe Institute of Technology, Karlsruhe, Germany

{simon.zeller,paul-niklas.kandora,niclas.kannengiesser,steffen.rebennack}@kit.edu

†Technical University of Munich, Campus Heilbronn, Germany

{daniel.kirste, sunyaev}@tum.de

Abstract—Concentrated liquidity automated market makers (AMMs), such as Uniswap v3, enable liquidity providers (LPs) to earn liquidity rewards by depositing tokens into liquidity pools. However, LPs often face significant financial losses driven by poorly selected liquidity provision intervals and high costs associated with frequent liquidity reallocation. To support LPs in achieving more profitable liquidity concentration, we developed a tractable stochastic optimization problem that can be used to compute optimal liquidity provision intervals for profitable liquidity provision. The developed problem accounts for the relationships between liquidity rewards, divergence loss, and reallocation costs. By formalizing optimal liquidity provision as a tractable stochastic optimization problem, we support a better understanding of the relationship between liquidity rewards, divergence loss, and reallocation costs. Moreover, the stochastic optimization problem offers a foundation for more profitable liquidity concentration.

Index Terms—automated market makers, cryptoeconomic systems, liquidity concentration optimization, liquidity provision, blockchain, stochastic optimization

I. INTRODUCTION

In decentralized exchanges (DEXs), automated market makers (AMMs) enable market participants to transparently trade assets represented as digital tokens [1]. Liquidity-concentrating liquidity provider (LP)–based AMMs, such as those used in Uniswap v3, are a specific type of AMM that sources tokens for trades from liquidity pools [2]. Liquidity pools are software components that store tokens deposited by LPs, such as individuals and organizations. In return for token deposits, LPs receive *liquidity rewards*—a share of the AMM's market-making revenue [1], [3]. Although gaining profits from liquidity provisioning may seem like a simple income source, most LPs suffer financial losses [4]–[6].

For profitable liquidity provisioning, LPs need to understand complex relationships between liquidity rewards, divergence loss, and reallocation costs. When opening a liquidity position by depositing tokens in a liquidity-concentrating LP-based AMM, LPs must carefully specify the liquidity provision interval for that position. Liquidity provision intervals are token price ranges in which an AMM can use an LP's deposited tokens for market making. If the token price falls within the provision intervals, the LP's liquidity position is active, and the LP gains liquidity rewards when the AMM uses their tokens for market making. If the token price moves outside the provision intervals, the AMM cannot use the tokens for market making. In this case, the LP's liquidity position becomes inactive, and the LP does not receive liquidity rewards.

LPs can increase liquidity rewards by concentrating liquidity in narrow liquidity provision intervals [2], [7]. Narrow liquidity position intervals, however, amplify divergence loss [8]. Divergence loss occurs when the value of tokens held by LPs in a liquidity pool diminishes due to price changes compared to the value they would have retained if the tokens were held outside the pool. Divergence loss decreases if the pool rebalances to the original token ratio when the LP made their deposit [3], [4].

Narrow liquidity provision intervals not only amplify divergence loss but also increase the risk of token prices moving beyond the specified range when compared to wider intervals [4]. To provide liquidity with an active position and gain liquidity rewards, LPs can reallocate their liquidity to a new interval around the token price. Liquidity reallocation, however, incurs reallocation costs, comprised of transaction costs charged by the blockchain system and rebalancing costs in the form of trading costs charged by the AMM.

To help LPs find optimal liquidity provision intervals to increase profits despite divergence loss and reallocation costs, extant research offers optimization approaches for liquidity provision intervals based on token volatility and token prices [6], [9], [10]. In these approaches, however, liquidity provision strategies are either modeled in continuous time, which deviates from practical implementations, potentially leading to suboptimal liquidity provision. Or, reallocation is treated as independent of the liquidity provision interval width, which may lead to the reallocation of an active liquidity position. To better account for reallocation costs when computing optimal liquidity provision intervals, we approach the following research question: *What is a useful approach to support LPs in finding optimal liquidity provision intervals to increase profitability?*

We first formalized liquidity concentration and the readjustment of liquidity provision intervals as a tractable stochastic optimization problem. Second, we used the optimization problem to devise an optimal liquidity reallocation strategy and applied the strategy to historical on-chain trading data of Uniswap v3. Comparing the devised strategy with other strategies showcases the utility of the tractable stochastic optimization problem to optimize liquidity provision intervals and their readjustments.

Our main ambition in this work is to support LPs in developing more profitable liquidity concentration strategies. By presenting a tractable stochastic optimization problem that optimizes the width of liquidity provision intervals and incorporates its influence on liquidity rewards, divergence loss, and reallocation costs, we support a better understanding of the trade-off between liquidity rewards, divergence loss, and reallocation costs. This contributes to increasing the profitability of liquidity provision and the efficiency of liquidity-concentrating AMMs.

The remainder of this work is structured into five sections. Section II explains the foundations of liquidity-concentrating AMMs relevant to this work and describes related research. Next, section III presents a formalization of profitable liquidity concentration in the form of an optimization problem. The utility of the formalization is then demonstrated in section IV. In section V, we discuss the principal findings and explain the main contributions of this work and its limitations. Moreover, we outline future research directions. Section VI concludes this work with a brief summary and key takeaways from this work.

II. BACKGROUND

This section first elucidates the foundations of liquidity-concentrating LP-based AMMs that are important in this work. Next, foundations of divergence loss and reallocation costs in liquidity concentration are explained.

A. Automated Market Makers

AMMs are software-based market makers that continuously quote bid (buy) and ask (sell) prices to exchange tokens with market participants in cryptoeconomic systems [1], [3]. To compute token prices, AMMs use mathematical functions implemented in software code of smart contracts [1]. Because smart contract code is visible to users with access to a blockchain system [11], pricing functions are transparent. AMMs use price discovery mechanisms to compute token exchange rates. This work focuses on AMMs with constant-product price discovery as a special type of AMM. These AMMs ensure that the product of the token quantities in a liquidity pool is invariant during trades [3], [12].

To provide liquidity to markets, AMMs require market participants to act as LPs by depositing tokens in liquidity pools [13]. To motivate market participants to deposit tokens, AMMs distribute fractions of collected market-making fees to LPs in the form of liquidity rewards. The amount of liquidity rewards is determined by the market-making fees collected and the proportion of an LP's liquidity relative to the total liquidity available to the AMM [3].

Basic AMMs like Uniswap v2 distribute provided liquidity uniformly across all possible token prices. This uniform distribution ensures that the same amount of liquidity is available for every possible token price. However, liquidity is primarily needed within a narrow range around the current token price. As a result, liquidity positioned far from this range remains largely unused, making uniform distribution inefficient [3].

To more efficiently provide liquidity, more advanced constant-product AMMs (e.g., Uniswap v3) enable LPs to concentrate provided liquidity by defining intervals for positions of deposited tokens [2], [14]. Liquidity concentration enables more efficient liquidity allocation, increasing the pool's liquidity within the LP's chosen intervals [2], [7]. AMMs only use tokens from active positions for market making. A position is active when the token price is within the liquidity provision intervals [2].

Liquidity concentration also benefits LPs. LPs can deposit tokens in narrow price intervals to amplify their share of liquidity rewards by increasing their share of provided liquidity within those intervals [4], [6]. While offering the chance to gain more rewards, narrow liquidity provision intervals increase the risks of divergence loss and opportunity costs, as inactive liquidity misses out on trading opportunities when prices move beyond the defined range [6]. Profitable liquidity provisioning thus requires careful selection and readjustment of liquidity provision intervals [6].

The adjustment of liquidity provision intervals entails reallocation costs, which comprise costs charged by the blockchain system for transaction processing and rebalancing costs charged by the AMM. Such costs can erode profits of LPs, especially those of LP's that provide lower amounts of tokens. For profitable liquidity provisioning, LPs must actively monitor market conditions and readjust their liquidity provision intervals to match token price developments.

B. Divergence Loss and Reallocation Costs in AMMs

Divergence loss occurs when token prices in a liquidity pool change after a deposit. For example, when a token A increases in price compared to a token B in the same liquidity pool, the token holdings in the liquidity pool change. This happens because the price reflects the ratio of tokens A and B. After this process, LPs own more of the lower-valued token B and less of the higher-valued token A. LPs withdrawing under these conditions suffer a financial loss compared to holding the deposited tokens outside of the liquidity pool. Divergence loss decreases if the liquidity pool recovers to the original token ratio from when the LP made their deposit [3], [4]. Narrower liquidity provision intervals are prone to higher divergence loss [8].

Compared to wider intervals, narrow liquidity provision intervals typically require more frequent liquidity reallocation to gain liquidity rewards. Reallocation incurs reallocation costs that comprise *transaction costs* and *rebalancing costs*. *Transaction costs* arise from fees that a blockchain system charges

for transaction processing. *Rebalancing costs* occur when LPs rebalance their token holdings for liquidity reallocation. In liquidity reallocation, LPs must first close an existing position and withdraw their tokens. Then, to open a new position, LPs must hold the tokens in a specific ratio, which often differs from the ratio of their withdrawn tokens. For example, if the price exceeds the upper limit of a liquidity provision interval, the LP will possess only token B. To open a new position with liquidity provision intervals around the token price, the LP must provide a specified quantity of both tokens A and B. Consequently, the LP needs to exchange tokens B for token A, incurring fees charged by the AMM. To own assets in that ratio, LPs rebalance their token holdings by exchanging tokens through the AMM's liquidity pool. However, using the AMM's liquidity pool incurs trading fees charged by the AMM. The fees are proportional to the amount of assets that the LP exchanges [2].

For profitable liquidity provisioning, LPs must find a Pareto-optimum regarding the trade-off between liquidity rewards, divergence losses, and reallocation costs [15], [16]. Narrow intervals increase liquidity rewards but increase the likelihood of the token price falling outside the interval, requiring more frequent reallocations and, therefore, higher reallocation costs. Token price movements additionally lead to divergence losses, which are amplified by narrower liquidity provision intervals and reduce overall returns for LPs. Conversely, broader intervals decrease liquidity rewards but also decrease divergence loss and the likelihood of the token price moving out of the liquidity provision intervals, thus decreasing reallocation costs. LPs must, therefore, appropriately balance the trade-off between liquidity rewards, divergence loss, and reallocation costs by selecting an optimal interval width.

C. Related Work on Optimal Liquidity Provision Strategies

Extant research has made valuable contributions to developing profitable liquidity provision strategies. Cartea et al. (2024) propose a mathematical model for optimal liquidity provisioning that can be used to compute optimal liquidity provision intervals based on price volatility, price drift, and pool profitability [6]. They present closed-form solutions based on a continuous-time model of liquidity provision with continuous reallocation. In practice, however, reallocation takes place at discrete points in time.

Fan et al. (2022) present profit and loss models for Uniswap v2 and v3, which solve sample average approximation (SAA) -based optimization problems but neglect liquidity reallocation [9]. Additionally, Fan et al. (2023) propose an approach to compute optimal liquidity allocation by maximizing SAA-approximated terminal wealth. They compute the optimal allocation of liquidity for a fixed reallocation rule that only depends on price movements. In their optimization approach, however, they do not optimize the interval width. Instead, they optimize the allocation of liquidity within a given interval of fixed size and therefore a fixed expected reallocation frequency.

III. PROFITABLE LIQUIDITY REALLOCATION AS AN OPTIMIZATION PROBLEM

This section first presents a mathematical description of liquidity-concentrating constant-product AMMs and reallocation costs. We use the mathematical description of constant-product AMMs and reallocation costs to represent the computation of liquidity provision intervals as a stochastic optimization problem, which we solve with commercial software. The optimization problem and its solution are envisioned to support LPs in developing more profitable liquidity provision strategies.

A. Mathematical Description of Constant Product Market Makers

Liquidity pools of constant-product AMMs represent reserves of token pairs (x, y). The total liquidity of a pool is denoted as $\overline{L}$ and calculated as $\overline{L} := \sqrt{xy}$. During token exchanges, the liquidity $\overline{L}$ of a pool is constant. An AMM exchanges Δ^x units of x for Δ^y units of y so that $\overline{L}$ remains constant: $\overline{L} = (x + \Delta^x, y - \Delta^y)$.

The marginal exchange rate y/x represents the price of one token x in terms of the other token y. We will simply refer to this as the price and use square root prices π to simplify further notation.

Uniswap v3 [2] is a special type of constant-product AMM that enables LPs to concentrate liquidity. In Uniswap v3, each LP must define liquidity provision intervals $[\pi^l, \pi^u]$ to open a liquidity position. If the token price falls within the interval of a position, the position is active, and the corresponding LP earns liquidity rewards proportional to its share of the total liquidity available in the pool for the token price [2].

Liquidity positions can be expressed through a triple (L, π^l, π^u), where L quantifies liquidity within the position's liquidity provision interval. Within $[\pi^l, \pi^u]$, deposited tokens are treated as *virtual reserves* to amplify the LP's provided liquidity in the liquidity provision interval. Uniswap v3 distinguishes between *real reserves* and *virtual reserves*. Real reserves refer to token amounts held by an LP, which we will denote in the following as (x^r, y^r). Virtual reserves represent liquidity concentrated within $[\pi^l, \pi^u]$. Real reserves associated with a liquidity position and square root price π can be computed with the following expression [15]:

$$(x^r, y^r) = \begin{cases} \left(L\left(\frac{1}{\pi^l} - \frac{1}{\pi^u}\right), 0\right) & \text{if } \pi < \pi^l, \\ \left(L\left(\frac{1}{\pi} - \frac{1}{\pi^u}\right), L\left(\pi - \pi^l\right)\right) & \text{if } \pi^l \le \pi \le \pi^l \\ \left(0, L\left(\pi^u - \pi^l\right)\right) & \text{if } \pi > \pi^u. \end{cases}$$
(1)

For (x^r, y^r), eq. (1) helps LPs figure out how much liquidity L they can provide based on π and a liquidity provision interval $[\pi^l, \pi^u]$. Before providing liquidity, the LP adjusts token amounts to be provided to ensure they are in the ratio satisfying eq. (1).

B. Readjustment and Cost Implications

LPs earn liquidity rewards proportional to their share of the pool's total liquidity, represented as $L/\overline{L}$. To increase liquidity

rewards, LPs concentrate their liquidity within narrow price ranges $[\pi^l, \pi^u]$, which increases L within those ranges. When the current π moves outside $[\pi^l, \pi^u]$, however, the liquidity position becomes inactive. LPs must reallocate their liquidity to earn liquidity rewards, which incurs reallocation costs (see section II-B). We denote transaction costs per reallocation as c^g and rebalancing costs per amount of exchanged tokens as c^{tr}.

C. Liquidity Provision as an Optimization Problem

We focus on a single LP to model the problem of optimal reallocation while considering liquidity rewards, divergence loss, and reallocation costs. The LP starts with initial wealth Y_0 given by the value of token holdings x_0^r and y_0^r. We assume their token holdings to be in the correct ratio for liquidity provision, i.e. $y_0^r = \pi_0^2 x_0^r$. The LP then uses all their wealth to provide liquidity from $t = 0$ to $t = T$. Given π_0^l and π_0^u, the initial tokens can be translated into initial liquidity L_0 using eq. (1) (and vice versa).

We assume the marginal price π_t^2 to follow a Geometric Brownian motion (GBM) process with constant volatility σ but without drift, as forecasting price trends is beyond the scope of this work. The optimal provision interval is expressed as $\left[\frac{1}{\alpha}\pi, \alpha\pi\right]$ for an optimal $\alpha > 1$ which is Pareto-optimal regarding the trade-off between liquidity rewards, divergence loss and reallocation costs [6]. In this work, α is independent of time.

In line with existing research [10], we assume the LP to reallocate their liquidity at time t when the liquidity position becomes inactive: $\pi_t \notin \left[\pi_{t-1}^l, \pi_{t-1}^u\right]$ As long as $\pi_t \in \left[\pi_{t-1}^l, \pi_{t-1}^u\right]$, the liquidity position remains unchanged to avoid reallocation costs. The proposed strategy, therefore, can be interpreted as an adapted version of a strategy that [10] refer to as *ULRA*.

Symbol	Description
x	First token
y	Second token
π_t	Square root of the marginal exchange rate
L_t	Liquidity provided by the LP
α	Relative size of the square root price interval
γ	Threshold that triggers reallocation
π_t^l	Lower bound of the LP positions square root price interval
π_t^u	Upper bound of the LP positions square root price interval
x_t^r	Real tokens x held by the LP
y_t^r	Real tokens y held by the LP
Y_t	Wealth of the LP
Σ_t^f	Accumulated and not-yet-claimed liquidity rewards
c^g	Transaction fees paid per interaction
c^{tr}	Fees paid by traders per unit of trade volume
$\overline{cf}_t$	Liquidity rewards paid to LP per time step and unit of provided liquidity, if their position is permanently active
c_t^f	Liquidity rewards paid to LP per time step and unit of provided liquidity
L_t	Liquidity provided by an LP
$\overline{L}_t$	Liquidity of the pool
T	Horizon for liquidity provision

TABLE I
LIST OF IMPORTANT SYMBOLS

To compute liquidity rewards, we calculate the proportion of time between $t - 1$ and t during which the position is active, meaning $\pi \in [\pi_{t-1}^l, \pi_{t-1}^u]$. The calculated proportion is based on linearly interpolated prices and is denoted as p_t. The proportion p_t is multiplied by $\overline{cf}_t$, which describes the liquidity rewards collected per unit of liquidity between $t - 1$ and t if the position were continuously active. We therefore express the liquidity rewards earned by the LP between $t - 1$ and t as $p_t \overline{cf}_t L_{t-1} = c_t^f L_{t-1}$. The LP does not change their active liquidity positions. The liquidity rewards are therefore collected until the LP reallocates their liquidity. This involves closing the existing position and reopening a new position using the accumulated liquidity rewards and the withdrawn tokens.

Terminal wealth (i.e., the value of token holdings together with the collected and not-yet-claimed liquidity rewards measured in USD) is maximized by determining the optimal liquidity provision interval size α. We use z_t as a binary variable that indicates whether or not the LP reallocates at time t. Due to the continuity of π_t, it does not matter whether we reallocate if $\pi_t \notin \left[\pi_{t-1}^l, \pi_{t-1}^u\right]$ or $\pi_t \notin \left(\pi_{t-1}^l, \pi_{t-1}^u\right)$. However, we use the second reallocation rule as it is easier to implement in an optimization problem. This leads to the following optimization problem:

$$\mathbf{P} : \max_{\alpha} \mathbb{E}\left[\mathbf{Y}_T\left(\pi_T, L_{T-1}, \pi_{T-1}^l, \pi_{T-1}^u, \Sigma_T^f\right)\right] \quad (2a)$$

$$\text{s.t.} \quad \pi_t \geq \pi_{t-1}^u \text{ or } \pi_t \leq \pi_{t-1}^l \Rightarrow z_t = 1 \text{ else } 0, \quad (2b)$$

$$\pi_t^l = z_t \cdot \frac{1}{\alpha}\pi_t + (1 - z_t) \cdot \pi_{t-1}^l, \quad (2c)$$

$$\pi_t^u = z_t \cdot \alpha\pi_t + (1 - z_t) \cdot \pi_{t-1}^u, \quad (2d)$$

$$\Sigma_t^f = \Sigma_{t-1}^f (1 - z_{t-1}) + c_t^f L_{t-1}, \quad (2e)$$

$$L_t = \mathbf{L}_t\left(L_{t-1}, \pi_{t-1}^l, \pi_{t-1}^u, \pi_t, \pi_t^l, \pi_t^u, \Sigma_t^f, c^{tr}, c^g\right), \quad (2f)$$

$$\alpha \geq 1 + \epsilon, \quad (2g)$$

$\mathbf{Y}_t$ and $\mathbf{L}_t$ refer to functions that map the parameters to Y_t and L_t (see eq. (1)). Equations (2b)-(2d) and (2f) must hold for $t = 1, \ldots, T - 1$ and eq. (2e) for $t = 1, \ldots, T$. We introduce $\epsilon > 0$ together with (2g) for numerical reasons and because it is not feasible to choose an arbitrarily small liquidity provision interval width within Uniswap v3 [2]. Formulation (2a)– (2g) constitutes a mixed-integer non-linear problem which can be solved using commercial software, such as Gurobi 12 [17]. For additional information concerning the tractable problem formulation of equations (2a)– (2g) and implementation details, we refer the reader to eq. (3) in section VI-A.

IV. UTILITY DEMONSTRATION

To illustrate the utility of the optimization problem for optimal liquidity provision, we analyzed on-chain data from Uniswap v3. We first analyze the profitability of LPs providing liquidity as described in section III-C for different values of α. Next, we compute an optimal solution to the stochastic

optimization problem using commercial software and validate the solution. In section IV-A, we describe the approach employed for the demonstration. The results are presented in section IV-B.

A. Methods

Following the modeling assumptions outlined in section III-C, we consider LPs with an initial token wealth $Y_0 = \pi_0^2 x_0^r + y_0^r$, providing liquidity in the manner described in the same section. Figure 1 illustrates the strategy. To further analyze the impact of the event triggering reallocation, we also formulate a modified strategy. In this strategy, liquidity is reallocated when $\pi_t < \pi_{t-1}^u - \gamma$ or $\pi_t > \pi_{t-1}^u + \gamma$. The LP therefore reallocates liquidity sooner when $\gamma < 0$, whereas it delays reallocation when $\gamma > 0$. For the demonstration, we used the
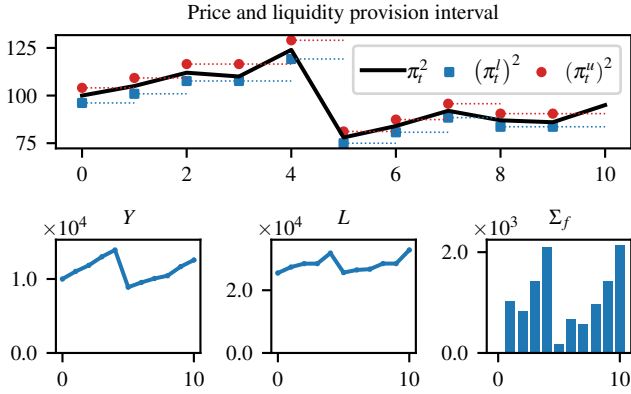

Fig. 1. Illustration of the evolution of an LP's liquidity provision interval, wealth, liquidity and unclaimed fees.

0.05% USDC/ETH liquidity pool on Uniswap v3—one of the most adopted pools by trade volume [18]. We consequently set c^{tr} to 0.05%. Our dataset spans from January 11, 2023, starting with the Ether price at approximately \$1,350, to November 30, 2024, with a price of \$3,725. The dataset is compiled from hourly data and contains the price together with *feeGrowthGlobal0X128* ($f_{g,0}$) and *feeGrowthGlobal1X128* ($f_{g,1}$). The variables $f_{g,0}$ and $f_{g,1}$ track the cumulative liquidity rewards that LPs would have earned from a position with liquidity 1 that has been active since the inception of the AMM [2]. Estimates for $\overline{c^f}_t$ are derived by calculating the changes in $f_{g,0}$ and $f_{g,1}$ between times $t-1$ and t and converting them into USDC. We consequently assume the LP to neither influence the price development nor the liquidity rewards earned per unit of liquidity ($f_{g,0}$ and $f_{g,1}$). Both variables are illustrated in fig. 2.

Similar to [9], we use SAA to estimate the objective function in eq. (2). This involves generating random price paths by simulating GBM, which are then handed to the formulation of the optimization problem described in the appendix (eq. (3)). As we used SAA to solve the optimization problem, the simulation is stochastic and not deterministic. To obtain reliable results, we solved the problem multiple times for different seeds.
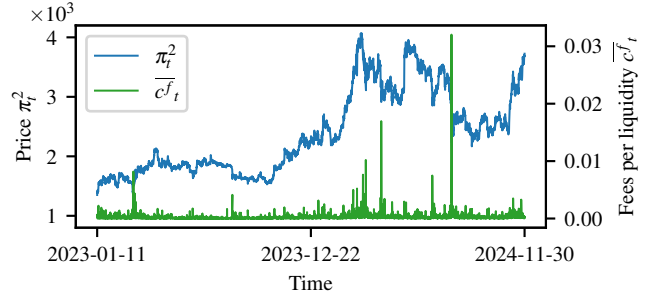

Fig. 2. Price and liquidity rewards per unit of virtual liquidity (L) during the analyzed time period.

The standard deviation of the GBM was estimated using the standard deviation of the log return of the price data and the drift was set to zero. Transaction costs were estimated to be \$ 109.8 per reallocation (\$ 84.8 as of [6] plus an additional estimated \$ 25 to claim liquidity rewards). The maximum liquidity rewards per liquidity $\overline{c^f}$ are set to the median of $\overline{c^f}_t$, and consequently assumed to be constant. The median was chosen as a robust estimator, as the data shows very large values of $\overline{c^f}_t$ at *price jumps*.

We bounded α to avoid numerical issues and to efficiently solve the optimization problem. The lower bound was set to 1.01 because values for α below ≈ 1.02 consistently resulted in profits of -100% in the used data set. We set the upper bound to 4, as values of α above 1.73 result in no reallocation taking place over the entire time frame of the used data. Additionally, already for $\alpha = 4$, the associated price interval covers movements ranging from $1 / 16$ to 16 times the initial price. The probability that any of the sampled price paths leaves the initial bounds is, therefore, very close to zero, given that the hourly σ of the GBM was estimated to ≈ 0.006 and paths are simulated for 5-10 steps. Reallocation costs therefore do not influence terminal wealth and the solution $\alpha = 4$ can be interpreted as going full range.

Given an initial wealth of the LP, the optimization problem is solved using a formulation of eq. (3) with SAA, that we solve with Gurobi [17]. The code can be accessed at https://github.com/simonczeller/AMM_Liquidity_Concentration_Optimization.

B. Demonstration Results

Figure 3 shows the profits an LP with an initial wealth of \$ 100,000 would have made if they provided their entire wealth in the Uniswap v3 0.05% pool for different liquidity provision interval sizes α from January 11, 2023 to November 30, 2024. The profits are shown for various liquidity provision interval sizes ranging from $\alpha = 1.0001$ to $\alpha = 4$. The corresponding average liquidity rewards and average reallocations per time step are shown in fig. 4 and fig. 5. Liquidity rewards (or reallocations) per time step are calculated as follows: If profits are greater than -100%, the total liquidity rewards (or reallocations) are divided by the total number of time steps in the analyzed data. If profits reach -100%, the denominator is given by the number of time steps until the LP has no
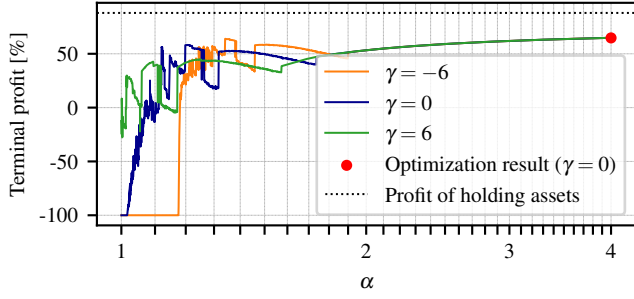
Fig. 3. Terminal profit of an LP with initial wealth $ 100,000 for different sizes α of the provision interval and values of γ, expressed as a percentage of the initial wealth.
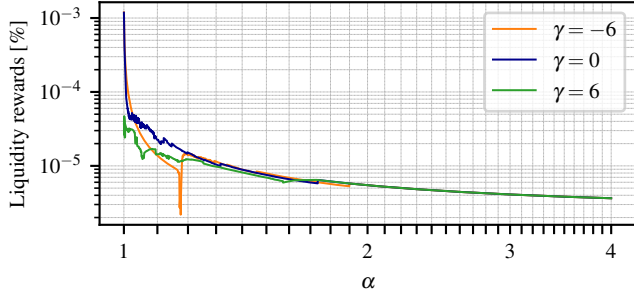

Fig. 4. Liquidity rewards per time step of an LP with initial wealth $ 100,000 for different sizes α of the provision interval and values of γ, expressed as a percentage of the initial wealth.

Fig. 5. Profits of an LP with initial wealth $ 100,000 for different sizes α of the provision interval and values of γ.

nitude of the price decrease and the specific path. A decrease (compared to an increase) would, however, not influence the number of reallocations, as they are mainly driven by volatility. Likewise, it would not affect the influence of α on DL which predominantly describes the profitability of larger positions. Consequently, the significant increase in the price of ETH in the analyzed period does not substantially bias the results.

Liquidity rewards were insufficient to offset LPs' divergence loss. The dotted line in fig. 3 illustrates the profits an LP would have gained by holding the original tokens–half of his wealth in USDC and half in ETH–rather than providing liquidity.

We solved the optimization problem (see eq. (2)) for an initial wealth of $ 100,000 and different combinations of T and number of sampled price paths S. Because the solution to the optimization problem is not deterministic as we use SAA, we have solved the problem multiple times to obtain reliable results. The values in Table II show that the optimal solution was consistently set to ≈ 4; equivalent to the value for α that an LP should have chosen in retrospect.

For $\gamma = 0$, the LP reallocates whenever $\pi_t \notin \left[\pi_{t-1}^l, \pi_{t-1}^u \right]$, despite a chance of the price reentering the interval. For this reason, it may be beneficial to reallocate only if π_t exceeds a specific distance to the interval. However, reallocating earlier reduces the probability of $\pi_{t+1} \notin \left[\pi_t^l, \pi_t^u \right]$ and therefore the expected time the position remains active, potentially increasing liquidity rewards. To evaluate such effects, the simulation was conducted for $\gamma = 6$ and $\gamma = -6$ and reallocation to be triggered if $\pi_t < \pi_{t-1}^u - \gamma$ or $\pi_t > \pi_{t-1}^u + \gamma$. For a negative value of γ, an increase of liquidity rewards is observed for very small α. However, the increase of reallocations with associated costs ultimately leads to less profitability for smaller values of α. A positive γ significantly decreased the number of reallocations, leading to a great increase in profitability for smaller values of α, despite less liquidity rewards.

There is insufficient data to determine whether varying values of γ affect profitability when the corresponding α is optimally chosen. The data, however, indicates that at least for small values of α a positive γ can increase the profitability of liquidity provision. Additionally, the optimization problem presented in eq. (2) and eq. (3) can be easily adapted to simultaneously optimize α and γ. A simultaneous optimization can be achieved by implementing γ as a decision variable and

remaining wealth. We will first discuss the results for $\gamma = 0$, to then examine the influence of a positive and negative value of γ. Figure 3 shows that for very small liquidity provision interval sizes α, despite very large liquidity rewards (fig. 4), the increased divergence loss together with the frequency of reallocations (fig. 5) with associated reallocation costs ultimately result in negative profits. For an increased α greater than ≈ 1.11, liquidity provision would have been profitable. Returns through liquidity returns and price increases therefore would have offset reallocation costs.

The jump in fig. 3 at $\alpha \approx 1.3$ is explained by only one reallocation taking place over the entire time-frame for $\alpha > 1.3$. This results in higher profits through decreased costs associated with reallocation. For $\alpha \geq 1.73$, the lower bound $\left(\pi_0^l \right)^2$ is smaller than the minimum price and the upper bound $\left(\pi_0^u \right)^2$ is greater than the maximum price. Thus, the initial liquidity provision interval is large enough such that no reallocation is needed (see fig. 5). The steady increase of profits for $\alpha \geq 1.73$ can be explained by the reduction of the divergence loss for larger liquidity position sizes ([8]). In this case, this effect outweighs lower liquidity rewards due to a decrease in virtual liquidity. Going full range would have resulted in profits of 71%, compared to 65% for $\alpha = 4$.

Liquidity provisioning is profitable in the analyzed period, mainly because the price of ETH nearly tripled. Conversely, a decrease in the price of ETH could have reduced profitability, potentially leading to financial losses depending on the mag-
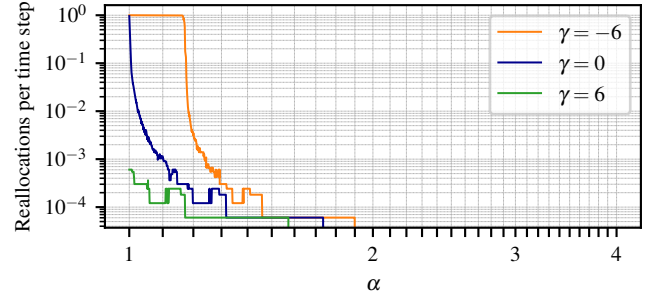
adapting the corresponding constraints.

V. DISCUSSION

This section presents the key findings and contributions of our research, examines major limitations, and explores potential strategies to address these limitations in future studies.

A. Principal Findings

The demonstration suggests that providing liquidity within wide liquidity provision intervals is profitable due to the significant increase in the marginal exchange rate. Due to the low virtual liquidity associated with such interval widths, liquidity rewards are too low to offset divergence loss. Providing liquidity in narrow intervals yields significantly higher liquidity rewards than wider intervals. Despite increased rewards, these strategies tend to generate lower profits or even lead to negative returns compared to using wider intervals. The reason is an increased divergence loss and more frequent reallocation, leading to higher reallocation costs. The data shows that LPs improve the profitability of strategies that use narrow liquidity provision intervals by delaying reallocation. LPs achieve this by only reallocating when the price distance to the interval exceeds a specified threshold, denoted as γ. This approach minimizes unnecessary reallocation costs while maintaining efficient liquidity management. Overall, even for an optimal α, the demonstration shows that liquidity provision would have been less profitable than not providing liquidity.

Moreover, the demonstration indicates that full-range liquidity provision intervals would have yielded the highest profitability. We attribute this outcome to the reduced divergence loss associated with wider intervals, combined with the price change observed during the analyzed time frame.

The demonstration shows that the formulation of the tractable stochastic optimization problem helps identify optimal provision intervals. The optimization problem takes the influence of the liquidity provision interval on liquidity rewards, divergence loss and reallocation costs into account and can be solved within a reasonable time using commercial software (see Table III). The computed optimal liquidity provision interval aligns with the interval an LP should have selected in hindsight. This highlights the model's potential to increase profits by optimizing liquidity rewards while accounting for divergence loss and reallocation costs.

B. Contributions

Our main ambition is to support LPs in developing more profitable liquidity provision strategies, including identifying and readjusting optimal liquidity provision intervals. We first formulate a liquidity provision strategy, that is straightforward to implement by LPs. Based on this strategy, a tractable stochastic optimization problem is formalized that seeks to optimize liquidity provision interval width. The optimization problem incorporates the three main components that determine the profitability of LPs: liquidity rewards, divergence loss, and reallocation costs. The formalization supports a better understanding of the relationships between the main components of liquidity provision profitability.

The stochastic optimization problem can be solved using commercial software. Moreover, it can easily be adapted to reflect various stochastic models of price movements and other market parameters, as well as user-defined thresholds for reallocation. This flexibility allows LPs to tailor the strategy to specific market conditions and provides a cornerstone for more profitable liquidity provision.

C. Limitations

To formulate liquidity provisioning as an optimization problem, we made multiple assumptions that limit the generalizability of our results. We assumed that LPs do not influence price movements and the total liquidity in the pool. While the influences on either of these may be negligible for small liquidity positions and larger pools, LPs with large liquidity positions providing liquidity in smaller pools may substantially influence total liquidity and token prices when rebalancing their token holdings.

We analyzed a finite-horizon problem with a limited number of time-steps T due to runtime constraints. However, the resulting optimal value for α may differ from the truly optimal value for an (infinite-horizon) problem or the maximization of wealth over more than T time steps.

Using the mathematical model, we calculated liquidity rewards by interpolating token prices. This implies that if $\pi_t \in \left[\pi_{t-1}^l, \pi_{t-1}^u\right]$, we assume constant liquidity rewards per unit of active liquidity between $t-1$ and t, i.e., $c_t^f = \overline{c^f}_t$. However, there is a nonzero probability that $\pi_s \notin \left[\pi_{t-1}^l, \pi_{t-1}^u\right]$ for some $s \in (t-1, t)$. This can lead to overestimating expected liquidity rewards, especially for very small values of α. An alternative to estimate the expected time the position is active between $t-1$ and t is to use Brownian bridges.

We assumed that the token price follows GBM without drift. Still, the optimization problem can easily be adapted to other stochastic processes, including GBM with drift and dynamic volatility, given that S price paths are simulated and handed to the optimization problem as parameters. Such an implementation may require adapting the liquidity provision interval, possibly by introducing time-dependent α_t or decision variables for the upper and lower bound of the liquidity provision interval.

D. Future Research

We tested the liquidity provision strategy based on on-chain data of Uniswap v3 based on discrete one-hour time steps. Additional empirical investigations are necessary to understand better how profitable the proposed strategy is when providing liquidity to other pools with different costs, rewards or price developments. Furthermore, the optimization approach presented in this work can be compared with other approaches (e.g., [6], [10]) on real-world data.

Many algorithms, such as Nested Benders Decomposition or Stochastic Dual Dynamic Programming [19], can reduce the runtime of stochastic optimization problems, enabling the

solution of large problems in acceptable time. Future research may adapt and implement the formulation of the optimization problem presented in this work for such algorithms to solve the problem for a larger sample size and more time steps.

The optimization approach relies on the accuracy of estimations of parameter values, including liquidity rewards per unit of liquidity, price volatility, and transaction fees. As forecasting those parameter values was beyond the scope of this work, accurate forecasting remains a potential area for future research.

VI. CONCLUSION

Liquidity concentration is complex because LPs must find Pareto-optima between liquidity rewards, divergence loss, and reallocation costs to increase profits from liquidity provision. This work presents a tractable stochastic optimization problem to assist LPs in computing optimal strategies for more profitable liquidity concentration—a foundational activity to the viability of liquidity-concentrating AMMs. The formulation of this problem can help LPs in finding the Pareto-optima between liquidity rewards, divergence loss, and reallocation costs by optimizing the interval width for liquidity concentration. Although liquidity concentration seems reasonable to increase the efficiency of using provided liquidity, this work shows that full-range liquidity positions (i.e., liquidity provisioning without liquidity concentration) are most profitable. We hope research and practice will leverage these findings to continue advancing the field of AMMs.

APPENDIX

A. Formulation for Optimization Software

The following presents a formulation that can be implemented and executed in optimization software.

$$\mathbf{P} : \max \mathbb{E}\left[\pi_T^2 \tilde{x}_T^r + \tilde{y}_T^r + \Sigma_T^f\right] \tag{3a}$$

$$\text{s.t.} \quad \alpha \geq 1 + \epsilon, \tag{3b}$$

$$\pi_t \leq \pi_{t-1}^l \Rightarrow z_t^l = 1 \text{ else } 0, \tag{3c}$$

$$\pi_t \geq \pi_{t-1}^u \Rightarrow z_t^u = 1 \text{ else } 0, \tag{3d}$$

$$z_t^m = 1 - z_t^l - z_t^u, \tag{3e}$$

$$z_t = z_t^l + z_t^u \tag{3f}$$

$$\pi_t^l = z_t \frac{1}{\alpha}\pi_t + (1 - z_t)\pi_{t-1}^l, \tag{3g}$$

$$\pi_t^u = z_t \alpha \pi_t + (1 - z_t)\pi_{t-1}^u, \tag{3h}$$

$$\tilde{x}_t^r = z_t^l L_{t-1}\left(\frac{1}{\pi_{t-1}^l} - \frac{1}{\pi_{t-1}^u}\right) + z_t^m L_{t-1}\left(\frac{1}{\pi_t} - \frac{1}{\pi_{t-1}^u}\right), \tag{3i}$$

$$\tilde{y}_t^r = z_t^m L_{t-1}\left(\pi_t - \pi_{t-1}^l\right) + z_t^u L_{t-1}\left(\pi_{t-1}^u - \pi_{t-1}^l\right), \tag{3j}$$

$$z_t\left(\pi_t^2 x_t^r + y_t^r\right) = z_t\left(\pi_t^2 \tilde{x}_t^r + \tilde{y}_t^r - c^{tr}\left|y_t^r - \tilde{y}_t^r\right| - c^g + \Sigma_t^f\right), \tag{3k}$$

$$z_t\left(y_t^r\right) = z_t\pi_t^2 x_t^r, \tag{3l}$$

$$(1 - z_t)x_t^r = (1 - z_t)\tilde{x}_t^r, \tag{3m}$$

$$(1 - z_t)y_t^r = (1 - z_t)\tilde{y}_t^r, \tag{3n}$$

$$L_t = z_t\left(\frac{\alpha}{\alpha - 1}\frac{y_t^r}{\pi_t}\right) + (1 - z_t)L_{t-1}, \tag{3o}$$

$$c_t^f = z_t^l \frac{\pi_{t-1}^2 - \left(\pi_{t-1}^l\right)^2}{\pi_{t-1}^2 - \pi_t^2}\overline{c^f} + z_t^m\overline{c^f} + z_t^u \frac{\left(\pi_{t-1}^u\right)^2 - \pi_{t-1}^2}{\pi_t^2 - \pi_{t-1}^2}\overline{c^f}, \tag{3p}$$

$$\Sigma_t^f = \Sigma_{t-1}^f(1 - z_{t-1}) + c_t^f L_{t-1}, \tag{3q}$$

$$z_t, z_t^l, z_t^m, z_t^u \in \{0, 1\}, \tag{3r}$$

$$\alpha, \pi_t^l, \pi_t^u, \tilde{x}_t^r, \tilde{y}_t^r, x_t^r, y_t^r, L_t, c_t^f, \Sigma_t^f \geq 0, \tag{3s}$$

B. Optimization Results and Runtime

The optimization results for different time steps T and sampling sizes S are show in table II. The computations were executed on a MacBook Air with M1 chip and 8 GB of RAM. The runtime for our optimization is shown in table III.

TABLE II
DESCRIPTIVE STATISTICS OF THE OPTIMAL SIZE α OVER 10 ITERATIONS FOR DIFFERENT TIME STEPS T AND SAMPLING SIZES S

T	5				10			
S	5	10	20	30	5	10	20	30
Mean	3.69	4.00	3.90	4.00	3.99	3.70	3.57	3.61
Median	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
Min	1.01	3.98	3.00	3.95	3.95	1.04	1.02	1.03
Max	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
Std Dev	0.94	0.01	0.32	0.01	0.02	0.93	0.93	0.95

TABLE III
DESCRIPTIVE STATISTICS OF THE RUNTIME IN SECONDS OVER 10 ITERATIONS FOR DIFFERENT TIME STEPS T AND SAMPLING SIZES S

T	5				10			
S	5	10	20	30	5	10	20	30
Mean	0.41	0.95	2.41	6.31	1.75	6.19	16.71	30.30
Median	0.30	0.76	1.99	5.58	1.73	5.47	13.86	32.27
Min	0.16	0.46	0.39	2.29	0.54	2.61	7.86	8.23
Max	0.94	2.09	5.00	13.50	3.43	15.44	33.18	51.33
Std Dev	0.27	0.54	1.47	3.30	0.94	3.84	9.17	12.27

REFERENCES

[1] D. Kirste, N. Kannengießer, R. Lamberty, and A. Sunyaev, "How automated market makers approach the thin market problem in cryptoeoconomic systems," 2023. [Online]. Available: https://arxiv.org/abs/2309.12818

[2] H. Adams, N. Zinsmeister, S. Moody, R. Keefer, and D. Robinson, "Uniswap v3 Core," 2021. [Online]. Available: https://uniswap.org/whitepaper-v3.pdf

[3] J. Xu, K. Paruch, S. Cousaert, and Y. Feng, "Sok: Decentralized exchanges (dex) with automated market maker (amm) protocols," *ACM Comput. Surv.*, vol. 55, no. 11, Feb. 2023. [Online]. Available: https://doi.org/10.1145/3570639

[4] J. Milionis, C. C. Moallemi, T. Roughgarden, and A. L. Zhang, "Automated Market Making and Loss-Versus-Rebalancing," Nov. 2023. [Online]. Available: http://arxiv.org/abs/2208.06046

[5] J. Aoyagi, "Liquidity Provision by Automated Market Makers," Rochester, NY, May 2020. [Online]. Available: https://papers.ssrn.com/abstract=3674178

[6] Á. Cartea, F. Drissi, and M. Monga, "Decentralized finance and automated market making: Predictable loss and optimal liquidity provision," *SIAM Journal on Financial Mathematics*, vol. 15, no. 3, pp. 931–959, 2024.

[7] M. Echenim, E. Gobet, and A.-C. Maurice, "Thorough mathematical modelling and analysis of Uniswap v3," Sep. 2023. [Online]. Available: https://hal.science/hal-04214315

[8] S. Hashemseresht and M. Pourpouneh, "Concentrated liquidity analysis in uniswap v3," in *Proceedings of the 2022 ACM CCS Workshop on Decentralized Finance and Security*, ser. DeFi'22, New York, NY, USA, 2022, p. 63–70. [Online]. Available: https://doi.org/10.1145/3560832.3563438

[9] Z. Fan, F. J. Marmolejo-Cossío, B. Altschuler, H. Sun, X. Wang, and D. Parkes, "Differential Liquidity Provision in Uniswap v3 and Implications for Contract Design," in *Proceedings of the Third ACM International Conference on AI in Finance*, ser. ICAIF '22, New York, NY, USA, 2022, pp. 9–17.

[10] Z. Fan, F. Marmolejo-Cossio, D. Moroz, M. Neuder, R. Rao, and D. C. Parkes, "Strategic Liquidity Provision in Uniswap V3," in *5th Conference on Advances in Financial Technologies (AFT 2023)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), J. Bonneau and S. M. Weinberg, Eds., vol. 282, Dagstuhl, Germany, 2023, pp. 25:1–25:22. [Online]. Available: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.AFT.2023.25

[11] N. Kannengiesser, S. Lins, C. Sander, K. Winter, H. Frey, and A. Sunyaev, "Challenges and Common Solutions in Smart Contract Development," *IEEE Transactions on Software Engineering*, vol. 48, no. 11, pp. 1–1, 2021.

[12] H. Adams, N. Zinsmeister, and D. Robinson, "Uniswap v2 Core," p. 10, 2020. [Online]. Available: https://uniswap.org/whitepaper.pdf

[13] D. Kirste, A. Poddey, and A. Sunyaev, "Undergirding Bonding Curves: Supply-sovereign Automated Market Makers Enabling Liquidity and Sustainable Financing," *Distrib. Ledger Technol.*, Feb. 2025, just Accepted. [Online]. Available: https://dl.acm.org/doi/10.1145/3716177

[14] S. Loesch, N. Hindman, M. B. Richardson, and N. Welch, "Impermanent loss in uniswap v3," 2021. [Online]. Available: https://arxiv.org/abs/2111.09192

[15] L. Heimbach, E. Schertenleib, and R. Wattenhofer, "Risks and returns of uniswap v3 liquidity providers," in *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, ser. AFT '22, New York, NY, USA, 2023, p. 89–101. [Online]. Available: https://doi.org/10.1145/3558535.3559772

[16] D. Kirste, A. Poddey, N. Kannengießer, and A. Sunyaev, "On the influence of conventional and automated market makers on market quality in cryptoeconomic systems," *Electronic Markets*, vol. 34, no. 1, p. 42, Aug. 2024.

[17] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2023, version 12.0. [Online]. Available: https://www.gurobi.com

[18] Uniswap, "Uniswap Interface," 2024. [Online]. Available: https://app.uniswap.org/positions

[19] C. Füllner and S. Rebennack, "Stochastic dual dynamic programming and its variants," 2021. [Online]. Available: https://optimization-online.org/?p=16920