Strategic Liquidity Provision in Uniswap v3

Michael Neuder
Harvard University
michaelneuder@g.harvard.edu

Rithvik Rao
Harvard University
rithvikrao@college.harvard.edu

Daniel J. Moroz
Harvard University
dmoroz@g.harvard.edu

David C. Parkes
Harvard University
parkes@eecs.harvard.edu

ABSTRACT

Uniswap is the largest decentralized exchange for digital currencies. The newest version, called Uniswap v3, allows liquidity providers to allocate liquidity to one or more closed intervals of the price of an asset, instead of over the total range of prices. While the price of the asset remains in that interval, the liquidity provider earns rewards proportionally to the amount of liquidity allocated. This induces the problem of *strategic liquidity provision*: smaller intervals result in higher concentration of liquidity and correspondingly larger rewards when the price remains in the interval, but with higher risk. We formalize this problem and study three classes of strategies for liquidity providers: uniform, proportional, and optimal (via a constrained optimization problem). We present experimental results based on the historical price data of Ethereum, which show that simple liquidity provision strategies can yield near-optimal utility and earn over 200x more than Uniswap v2 liquidity provision.

CCS CONCEPTS

• **Applied computing** → **Electronic commerce**; **Digital cash**; **E-commerce infrastructure**; **Secure online transactions**; **Electronic funds transfer**.

KEYWORDS

blockchain, decentralized finance, Uniswap v3, liquidity provision

1 INTRODUCTION

Decentralized finance (DeFi) is a large and rapidly growing collection of projects in the cryptocurrency and blockchain ecosystem. From May 2020 to May 2021, the TVL (total value locked, meaning the sum of all liquidity provided to the protocol) into DeFi protocols ballooned by 100x from 800 million USD to 80 billion USD [15], beginning with a period of rapid growth in the "DeFi Summer" of 2020. DeFi aims to replicate traditional financial intermediaries and instruments using smart contracts executed on blockchains (typically Ethereum).

Decentralized exchanges (DEXes) allow users to swap tokens of different types without a trusted intermediary. Most DEXes, including Uniswap, fall into the category of *constant function market makers* (CFMMs). Instead of using an order book as is done in traditional exchanges, CFMMs use an *automated market maker* (AMM) to determine the price of an asset. In *Uniswap v2*, token pairs can be swapped for each other using *liquidity pools*, which contain both tokens. Permitted trades are determined by the *reserve curve*, $xy = k$, where x and y denote the the number of tokens of each type in the liquidity pool, and k remains constant across trades. *Liquidity providers* add tokens to liquidity pools for the traders to swap against, and are rewarded through the fees traders pay. An illustrative reserve curve is shown in Figure 1 (blue). In order to trade some quantity of token y for some quantity of token x, the trader must keep the product of reserves constant, i.e., buy Δx of x for Δy of y such that $(x - \Delta x)(y + \Delta y) = k$.

This reserve curve also defines an effective price for token x in units of token y, i.e., $p_x(x, y) = -dy/dx$ [12]. In the context of the $xy = k$ curve of Uniswap v2, we have $y = k/x \implies p_x(x, y) = k/x^2$. By convention, we take the "price" corresponding to the AMM and liquidity pool to be the price of x, i.e., $p_x(x, y)$, and we let the x token be volatile relative to the y token. In Uniswap v2, liquidity providers earn rewards when traders use the liquidity to execute swaps, which each incur a constant fee of 0.3% [2]. Each liquidity provider provides liquidity on the entire interval of possible prices $(0, \infty)$, and earns rewards based on their fraction of the total liquidity in the pool.[1]

On May 3, 2021, Uniswap's new protocol, dubbed *Uniswap v3* [3], was launched on the Ethereum mainnet. The primary update of Uniswap v3 over Uniswap v2 was the addition of *concentrated liquidity* [3]. Within three weeks, the new protocol accumulated a TVL of over 1.2 billion USD and averaged a daily volume of 1.6 billion USD [17]. In Uniswap v3, liquidity providers can provide liquidity to each of any number of price intervals, called *positions*. Liquidity allocated to position $[p_a, p_b]$ earns rewards from fees when the price remains in that interval. If multiple providers have allocated liquidity over an interval containing the correct price, each is rewarded proportionally to the fraction of the liquidity they own on that price. Figure 1 (red) demonstrates how the constant product curve of Uniswap v2 is shifted to intercept the axes at a and b, which are determined by the upper and lower bound of the price interval of the position. This shifted curve [3] is given by

$$\left(x + \sqrt{k/p_b}\right)\left(y + \sqrt{kp_a}\right) = k, \tag{1}$$

and the intercepts, a and b can be calculated by letting x or y equal zero respectively.

In this way, Uniswap v3 supports a diversity of strategies in regard to the allocation of liquidity. Each liquidity provider is presented with a trade off between choosing large positions that cover

Submitted, June 2021, arXiv version

[1] The end behavior of this derivative demonstrates the support for the range of prices being the interval $(0, \infty)$, with $\lim_{x\to\infty} k/x^2 = 0$, and $\lim_{x\to 0^+} k/x^2 = \infty$.

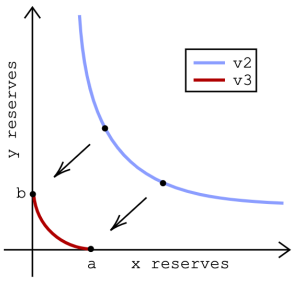

Figure 1: The reserve curves of Uniswap v2 and v3. Providing concentrated liquidity in v3 on price interval $[p_a, p_b]$ results in shifting the Uniswap v2 curve $xy = k$ to intercept the axes at a and b respectively. The intercepts are calculated by setting x and y equal to zero in the v3 reserve curve (Equation 1).

many possible prices, but earn less fees than smaller, more concentrated positions that are also more risky. Additionally, there is a cost of reallocating liquidity, because liquidity allocations are transactions that must be included in a block and thus incur gas fees. This cost must be factored into a liquidity provider's strategy.

The contributions of this paper are as follows:

(1) formalizing the strategic liquidity provision problem and a family of liquidity provision strategies which we call "reset liquidity provision strategies" (reset-LP strategies),
(2) presenting three classes of reset-LP strategies for liquidity providers, which we call *uniform*, *proportional*, and *optimal*,
(3) analytically calculating the expected utility of a reset-LP strategy,
(4) solving for the optimal reset-LP strategy based on price-change statistics created from historical Ethereum price data,
(5) demonstrating that proportional allocations are optimal for risk-seeking providers and uniform allocations are optimal for risk-averse providers, and
(6) back-testing an optimal reset-LP strategy to demonstrate that under suitable conditions, a strategic provider will earn 200x more return on investment than through following a v2 strategy.

1.1 Related work

The Uniswap v1 protocol was defined by Adams [1], followed up with v2 by Adams et al. [2], and most recently amended in v3 by Adams et al. [3].

There is a growing body of work studying liquidity provision incentives in Uniswap v2. Angeris et al. [6] present an analysis of Uniswap, and more broadly of constant-product AMMs, and demonstrate conditions for which the markets closely track the price on an external reference market. Angeris and Chitra [4] extend this line of research by demonstrating that the more general class of CFMMs incentivize participants to report the true price of an asset on an external market, demonstrating their value as price oracles.

Evans [10] demonstrates that for *geometric mean market makers* (G3Ms), passive liquidity provision can be used to replicate payoffs of financial derivatives and more active trading strategies. Tassy and White [16] and White et al. [18] analyze the growth in wealth of a liquidity provider in CFMM for a geometric Brownian motion price process. Evans et al. [11] extend this to more general liquidity provider objectives and diffusion processes.

Aoyagi [7] study the equilibrium liquidity provision of constant product markets, and show that strategic liquidity providers in the Uniswap v2 environment may have a non-monotonic best response when parameterized with the opponents' liquidity provision. Angeris et al. [5] extend this line of work to arbitrary CFMMs and calculate bounds on the liquidity provider rewards based on the curve that defines the CFMM.

The above work applies to Uniswap v2 and the general class of CFMMs but not to Uniswap v3. A blog post by Max [13] describes a "passive rebalancing" strategy for v3, which aims at maintaining a 50-50 ratio of value for the assets of the liquidity provider. Our work presents a more general class of liquidity provision strategies for Uniswap v3, and introduces a Markov model to evaluate the expected utility of different strategies. To our knowledge, this is the first formal study of liquidity provision strategies in Uniswap v3.

1.2 Outline

Section 2 introduces the Uniswap v3 protocol and introduces the concept of a liquidity provision strategy. We primarily focus on a class of strategies which we refer to as "τ-reset" strategies. Section 3 presents the Markov model used to analyze the expected utility of a strategy in this class of strategies. Section 4 presents three specific liquidity provision strategies including the optimal τ-reset strategy. Section 5 presents empirical results based on historical Ethereum price data. Section 6 suggests open problems for future research and concludes.

2 UNISWAP V3

Uniswap v3 introduces to AMMs the concept of *concentrated liquidity*. Instead of providing liquidity on the entire price range of $(0, \infty)$, providers can now specify one or more intervals of price for one of the assets over which to provide liquidity. A provider earns transaction fees when the price of the specified asset is within one of these intervals (and only then). Additionally, if multiple providers allocate liquidity to the same price, they are each rewarded proportionally to the fraction of the total liquidity of that price that they own.

By choosing more concentrated intervals, a provider can increase their return when the price remains in the interval, but this will also increase the variance in their return. To formalize this, we model a discrete set of price bins, and a provider who chooses how much liquidity to place in each bin and when to reallocate, and who seeks to maximize its expected utility from the stream of fees implied by the adopted strategy.

Definition 2.1 (Bins). We define a set of *bins* $B = \{b_1, b_2, \ldots, b_c, \ldots\}$, with each bin b_i corresponding to price interval $[l_i, r_i)$, and these form a partition of $[0, \infty)$, with $l_1 = 0$ and $r_i = l_{i+1}$ for all $i \in \{1, 2, \ldots\}$. Bin b_i corresponds to interval $[l_i, r_i)$. Bin b_c denotes the bin containing the *current price* of the asset.

For the remainder of this work we refer to the price of one asset in a token pair as measured in units of the other. For example, the

USDC/ETH pool, where we measure the volatile price of ETH in the stable units of USDC. Consider time $t = 0$ and let b_t denote the bin that contains the current price of the volatile asset. A liquidity provision strategy at time $t = 0$ provides a method of determining what proportion of the provider's liquidity is allocated to each bin. We make the following assumptions:

(1) **Stable price distribution.** We assume the next-price distribution, describing the percent change in price relative to the current price, is constant across time and invariant to the current price. We validated this empirically using the Ethereum 10-minute historical price data, where we found a correlation coe cient of $R^2 = 0.98$ between the following pairs of probability distributions (i) ETH prices above 300 USD against ETH prices below 300 USD (ii) ETH prices from April 2018-April 2019 against ETH prices from April 2019-April 2020.
(2) **Fixed cost to reallocate liquidity.** We assume that the cost to reallocate the liquidity, which comes from the gas fee of including the associated transaction in a block, is xed (xed to 1), with other values normalized relative to this cost. For example if the liquidity provider allocates $L = 100$ units of liquidity, this is interpreted as 100 times the cost of reallocating liquidity.
(3) **Periodic updates.** We assume that a provider's liquidity allocation is updated periodically, and with an immediate e ect of any reallocation. Further, we take the period length to be long enough (at least 10 minutes) that network transmission delay is not a factor, and this is not the focus of this work.
(4) **Single strategic provider.** We assume a single, strategic liquidity provider, and implicitly model the rest of the providers as allocating liquidity over the entire price range, i.e., following the Uniswap v2 liquidity provision method.

2.1 Liquidity provision strategies

In describing the strategic liquidity provision problem, we rst de ne the stochastic process $\{b_t : t = 2 \mathbb{N}\}$ on the price b_t at time index t. We model a stable next-price distribution, such that the distribution on next price, describing the change in price relative to the current price, is constant across time and also invariant to the current price. For this, we re-index the price bins relative to the current price. Let b_0 denote the current price bin, and index this in relative terms as b_0. Let b_{-1} and b_{+1} denote the -1 bin to the left and right b_0, respectively. The support of the next-price distribution is within set $\Delta = \{-\Delta_{max}, -\Delta_{max}+1, \ldots, 0, \ldots, \Delta_{max}-1, \Delta_{max}\}$, where Δ_{max} is the maximum possible next-price movement. Following Assumption 1, we can write

$$\Pr[b_{t+1} = b_{0+\delta} \mid b_t = b_0] = \alpha_\delta \quad \text{for } \delta \in \Delta \tag{2}$$

where α_δ is the probability of moving left or right by δ bins, depending on the sign of δ.

Given this, we can now de ne a simple class of liquidity provision strategies.

De nition 2.2. A reset liquidity provision strategy (reset-LP strategy) is composed of:

(1) The bin containing the price at the time of reset, $b_0 = b_0$.
(2) An allocation, $\{\ell_\delta : \delta \in \Delta\}$, that speci es the fraction of liquidity allocated to each bin b_δ in Δ.
(3) A reset condition, which speci es a subset of the bins in that cause the strategy to reset. Upon a reset, the allocation rule is used to reallocate liquidity, centered on the new price b_0.

Of particular interest is the family of g-reset strategies.

De nition 2.3. A g-reset strategy is a reset-LP strategy in which the reset condition is de ned so that there is a reset only when the price is outside the set $g = \{b_{-\delta_g}, \ldots, b_0, \ldots, b_{+\delta_g}\}$ of $2\delta_g + 1$ of contiguous bins, centered on b_0.

Sometimes we also use g to denote the probability mass of the next-price distribution that is covered by g. For example, if $g = 0.50$ then δ_g is selected to be the smallest number such that set g includes at least 50% of the next-price probability mass.

We also sometimes write g to denote the set of relative indices corresponding to this set of bins, i.e., $g = \{-\delta_g, \ldots, 0, \ldots, \delta_g\}$. The usage will be clear from the context.

For illustration, consider the following strategies.

> **Example 1 (Fixed strategy)** Always provide liquidity in the price interval [\$30, \$50].
>
> **Example 2 (Uniform g-reset strategy)** Allocate liquidity uniformly on a range of bins centered on the current price b_0. Reset when the price moves outside this range."
>
> **Example 3 (Proportional g-reset strategy #1)** Let $g = 0.5$, so g contains the middle 50% of the probability mass of the next-price distribution. Allocate liquidity proportionally to the probability of each bin in g. Reset according to g."
>
> **Example 4 (Proportional g-reset strategy #2)** Let $g = 0.5$, so g contains the middle 50% of the probability mass of the next-price distribution. Allocate liquidity proportionally to the probability of each bin in the middle 90% of the probability mass of the next-price distribution. Reset according to g."

The uniform g-reset strategy is illustrated in Figure 2.

Figure 2: The uniform g-reset strategy, de ned here on three contiguous bins centered on the current price. Each circle represents a price bin, and the darker circle delineates the current price at each time step. Once the price leaves these three contiguous bins, the strategy "resets" and re-centers the allocation to the price at the time of the reset.

3 MARKOV ANALYSIS

In this section, We develop an analytical framework with which to determine the expected utility from the fees that ow from a particular reset-LP strategy.

De nition 3.1. Using relative indices, let $\mathbb{P}(\cdot,\cdot) : \mathbb{Z} \times \mathbb{Z} \to [0,1]$ denote the probability of the price moving from the i^{th} bin in t to the j^{th} bin in $t+1$,

$$\mathbb{P}(i,j) = \Pr[p_{t+1} = b_j \mid p_t = b_i] \quad \text{for } i,j \in \mathbb{Z} \tag{3}$$

Note that $\mathbb{P}(i,j) = \mathbb{P}(0, j-i)$, because de nes the probability that the price moves $j-i$ bins.

Given $p_t = b_i$, there is also a non-zero probability that the next price p_{t+1} lands in a bin b_i $i \neq g$. This is the complement of the probability of landing in any of the bins b_j $j \neq g$.

De nition 3.2. Let $\mathbb{R}(\cdot) : g \to [0,1]$ denote the probability of resetting, given the price is in the i^{th} bin of g,

$$\mathbb{R}(i) = 1 - \sum_{j \in g} \mathbb{P}(i,j) \quad \text{for } i \in g \tag{4}$$

Landing in a bin outside of g causes a reallocation of liquidity that is centered on b_0 (by the de nition of the g-reset strategy), so we can also view a reset as transitioning the price to the b_0 bin.

3.1 Reset Markov chain

De nition 3.3. The reset Markov chain, $\Gamma \in \mathbb{R}^{|g| \times |g|}$, where $\Gamma(i,j)$ denotes the probability from transitioning to state j given the current state is i, describes the stochastic process of the price movement over the set of bins g, induced by a g-reset strategy. We have:

$$\Gamma = \begin{bmatrix} \mathbb{P}(0,0) & \mathbb{P}(0,1) & \cdots & \mathbb{P}(0,|g|-1) \\ \mathbb{P}(1,0) & \mathbb{P}(1,1) & \cdots & \mathbb{P}(1,|g|-1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{P}(|g|-1,0) & \mathbb{P}(|g|-1,1) & \cdots & \mathbb{P}(|g|-1,|g|-1) \end{bmatrix}$$

De nition 3.4. Given g, let $\gamma_g \in \mathbb{R}^{1 \times |g|}$ denote the stationary distribution of the reset Markov chain, where

$$\gamma_g \Gamma = \gamma_g \tag{5}$$

At any time, the probability that the price is contained within the i^{th} bin of g, given a g-reset strategy, is $\gamma_g(i)$.

3.2 Liquidity allocation

While playing a g-reset liquidity provision strategy, agents may want to allocate liquidity to a larger set of bins than g.

De nition 3.5. Let the U bins, $b_{-U}, \ldots, b_U$, denote a set of bins $2U+1$, centered at b_0, over which liquidity is allocated.

Note that,

$$U = \{b_{-U}, \ldots, b_0, \ldots, b_U\} \tag{6}$$

Similarly to g we also overload U to denote the set of relative indices corresponding to this set of bins,

$$U = \{-U, \ldots, 0, \ldots, U\} \tag{7}$$

and the usage will be clear from context.

De nition 3.6. A liquidity allocation, $\lambda : U \to [0,1]$ de nes the fraction of a provider's liquidity allocated to each bin in U, and satis es $\sum_{i \in U} \lambda(i) = 1$.

In order to evaluate this allocation, for each $j \in U$, we need (i) the probability of being in bin b_j at any given time, and (ii) the utility of being in bin b_j.

De nition 3.7. The outcome matrix, $\Psi \in \mathbb{R}^{|g| \times |U|}$ is,

$$\Psi = \begin{bmatrix} \mathbb{P}(b_0, b_{-U}) & \mathbb{P}(b_0, b_{-U+1}) & \cdots & \mathbb{P}(b_0, b_U) \\ \mathbb{P}(b_1, b_{-U}) & \mathbb{P}(b_1, b_{-U+1}) & \cdots & \mathbb{P}(b_1, b_U) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{P}(b_{|g|-1}, b_{-U}) & \mathbb{P}(b_{|g|-1}, b_{-U+1}) & \cdots & \mathbb{P}(b_{|g|-1}, b_U) \end{bmatrix} \tag{8}$$

which represents the probability of moving from any bin in g to any bin in U.

The probability of landing in the j^{th} bin of U is calculated by marginalizing out the current location (taking the dot product of the stationary distribution of the Markov chain with each column of the outcome matrix),

$$\Pr[p_{t+1} = b_j] = \sum_{i \in g} \gamma_g(i)\, \mathbb{P}(i,j) \quad \text{for } j \in U \tag{9}$$

To de ne the utility of each bin in U, we use the exponential (constant absolute risk aversion) utility function [8, 14].

De nition 3.8. Let $\mathbb{U}(\cdot)$ denote the exponential utility function,

$$\mathbb{U}(x) = \begin{cases} \frac{1 - e^{-\alpha x}}{\alpha} & \alpha > 0 \\ x & \alpha = 0 \end{cases} \tag{10}$$

where $\alpha \in \mathbb{R}$ denotes the Arrow Pratt measure of absolute risk aversion, a proxy for the provider's degree of risk preference. Risk-seeking, risk-averse, and risk-neutral preferences correspond to $\alpha < 0$, $\alpha > 0$, and $\alpha = 0$ respectively.

De nition 3.9. The reward, $\phi(j)$, of landing in the j^{th} bin of U is proportional to the amount of liquidity allocated to that bin. Let $\hat{\lambda} \in \mathbb{R}$ denote the proportionality constant, which is the fraction of the amount of liquidity provided that is earned as a reward in the next time step. From Assumption 4, we assume that $\hat{\lambda}$ is constant for each central bin $b_i \in$ and at each time $t = .$

Additionally, the reward incurs a reset fee of $1-\gamma$ if the bin is not in g, and we have reward function:

$$\phi(j) = \begin{cases} \hat{\lambda}\, \lambda(j) & \text{for } j \in g \cap U \\ \hat{\lambda}\, \lambda(j)\,(1-\gamma) & \text{for } j \notin g \cap U \end{cases} \tag{11}$$

The utility will depend on the risk preference of the provider. Here, we increment each reward by one unit, making the exponential utility function (Equation 10) well de ned (it requires positive inputs). This constant shift does not a ect the comparison of expected utility across di erent strategies.

De nition 3.10. The utility, $\mathcal{U}(\cdot)$, of landing in the j^{th} bin of U is

$$\mathcal{U}(j) = \mathbb{U}(\phi(j) + 1) \quad \text{for } j \in U \tag{12}$$

Figure 3: The price dynamics in the illustrative example. At each time step, there is a one-third probability of moving left one, moving right one, and staying in the same bin.

Figure 4: The reset Markov chain in the illustrative example. The center node will move left, right, or stay in the same location with probability one-third, which means it cannot cause a reset, and thus will always earn a reward of , ^ •3. The outer nodes will cause a reset with probability one-third, which will incur a reward of 1.

De nition 3.11. The expected utility $_D$, of an allocation, , is

$$_D{}^1 \ ^o = \tilde{O}_{92\ U} \Pr\ \%_{\text{¸}, 1} = 1_{1^9} \ * \ {}^1 9^{\text{”}} \qquad (13)$$

3.3 Illustrative example

Consider the simple next-price distribution,

$$\Pr{}^1\%_{\text{¸}, 1} = 1_{1\ 1^o} \ j \ \%_{\text{¸}} = 1_{1 0^o} {}^o = 1 \bullet 3$$
$$\Pr{}^1\%_{\text{¸}, 1} = 1_{1 0^o} \ j \ \%_{\text{¸}} = 1_{1 0^o} {}^o = 1 \bullet 3$$
$$\Pr{}^1\%_{\text{¸}, 1} = 1_{1 1^o} \ j \ \%_{\text{¸}} = 1_{1 0^o} {}^o = 1 \bullet 3\text{”}$$

That is, the price will move down one bin, up one bin, and stay in the same bin, each with probability one-third. This price model is represented in the Markov chain depicted in Figure 3. We use the following U and g bins,

$$_g = \ _U = f 1_{1\ 1^o} \bullet 1_{1 0^o} \bullet 1_{1 1^o} g \bullet$$

which also implies that = $_U$ = $_g$ = 1. We now can calculate the reset Markov chain for the example,

$$\text{"} = \begin{matrix} 2 & & & 3 \\ 6 & 1 \bullet 3 & 2 \bullet 3 & 0 \ 7 \\ 6 & 1 \bullet 3 & 1 \bullet 3 & 1 \bullet 3 7 \\ 6 & & & 7 \\ 4 & 0 & 2 \bullet 3 & 1 \bullet 3 5 \end{matrix} \text{”}$$

Figure 4 shows this Markov chain visually, where the blue arrows are non-reset transitions, and the red arrows are reset transitions. The stationary distribution of this Markov chain is given by the left eigenvector with eigenvalue 1,

$$?_g = \ 1 \bullet 4 \quad 1 \bullet 2 \quad 1 \bullet 4 \ \text{”}$$

This means that at any time step, there is a one-half probability that the current price is in the middle bin of the strategy, and there is a one-fourth probability that the current price is in one of the edge bins of the strategy. Let the example allocation be,

$$ {}^1 9^{\text{p}} = 1 \bullet 3 \bullet \quad \text{for } 9 2 \ _U \bullet$$

so each bin in $_U$ is allocated one-third of the provider's liquidity. We have the outcome matrix $\$$,

$$\$ = \begin{matrix} 2 & & & 3 \\ 6 & 1 \bullet 3 & 1 \bullet 3 & 0 \ 7 \\ 6 & 1 \bullet 3 & 1 \bullet 3 & 1 \bullet 3 7 \\ 6 & & & 7 \\ 4 & 0 & 1 \bullet 3 & 1 \bullet 3 5 \end{matrix} \text{”}$$

Let = ^ = 1. By taking the dot product of the stationary distribution, $?_g$ with each column of the outcome matrix, we calculate the expected utility. If we let 0 = 0 (risk-neutral), then,

$$_D{}^1 \ ^o = 1 \bullet 3 \ \ 5 \bullet 6 \ , \ 0 \ \ 1 \bullet 6$$
$$= 5 \bullet 18 = 0 \text{”} 27 \text{”}$$

At each time step, we can expect to earn 5•18 reward for each unit of liquidity that we provide.

4 LIQUIDITY PROVISION STRATEGIES

We now present three g-reset strategies.

4.1 Proportional strategy

In this strategy, the provider allocates liquidity proportionally to the probability of landing in a certain U bin.

De nition 4.1. The proportional strategy is a g-reset strategy with:

(1) The bin, 1_B, of the price when re-setting the strategy.
(2) The smallest set of consecutive bins, $_g$, centered at 1_B, which account for at least g of the probability mass of the next-price distribution.
(3) The smallest set of consecutive bins, $_U$, centered at 1_B, which account for at least U of the probability mass of the next-price distribution.
(4) The allocation function

$$ {}^1 9^{\text{p}} / \ {}^1 9^{\text{p}} \bullet \quad \text{for } 9 2 \ _U \bullet \qquad (14)$$

which allocates all of the liquidity proportionally to the probability of landing in each bin.

Figure 5 shows an example of the proportional strategy, for the case of U ¡ g. If U Ÿ g, the set of g bins would be larger than the set of U bins.

4.2 Uniform allocation strategy

In this strategy, the provider allocates liquidity uniformly over the set of U bins.

De nition 4.2. The uniform allocation strategy is a g-reset strategy with:

Figure 5: An example of the proportional g-reset strategy, where U ¡ g. The height of the bars indicates the amount of liquidity in each bin. When the strategy last reset, the price was 1B. The next-price probability distribution is shown in blue. The alpha" and the tau" bins are illustrated. In this case, the middle ve bins are part of both U and g.

(1) The bin 1B of the price when resetting the strategy.
(2) A set of consecutive bins g .
(3) A set of consecutive bins U .
(4) The allocation function

$$\ell_9 = 1 \cdot 1 2 = U \, , \, 1^{o} \bullet \quad \text{for } 9 2 \; U \bullet \tag{15}$$

where =U is the number of bins in U.

4.3 Optimal liquidity strategy

In this strategy, the provider allocates liquidity optimally (among g-reset strategies) over the set U of bins for a speci ed set of consecutive bins, g

De nition 4.3. The optimal liquidity strategy is de ned by:

(1) The bin 1B of the price when resetting the strategy.
(2) A set of consecutive bins g .
(3) A set of consecutive bins, U , de ned to contain every bin that could be transitioned to from a bin in g.
(4) An allocation function, , that is the solution to the liquidity optimization problem, de ned as

$$\begin{aligned} \max_{2R^{U}} \quad & D^{1} \; o = \tilde{O} \Pr \; \%_{e \, , \, 1} = 1_{1 9^{o}} \; * \; \ell_9 \\ & 9 2 \; U \\ \text{s.t.} \quad & \tilde{O} \; \ell_9 = 1 \\ & 9 2 \; U \\ & \ell_9 \quad 0 \quad \text{for } 9 2 \; U \text{”} \end{aligned} \tag{16}$$

The constraints specify that (i) all of the liquidity is allocated, and (ii) the liquidity allocated to each bin is non-negative.

If an interior solution exists, this optimization problem admits a standard solution by way of Lagrange multipliers. Then the solution is characterized by the system

$$* \; {}^{0}{}_{1 9^{o}} \; \Pr \; \%_{e \, , \, 1} = 1_{1 9^{o}} \; = * \; {}^{0}{}_{1 : \; o} \; \Pr \; \%_{e \, , \, 1} = 1_{1 : \; o}$$

for all 9 • : 2 U, and the constraints

$$\tilde{O} \; \ell_9 = 1$$
$$9 2 \; U$$

and

$$\ell_9 \quad 0 \quad \text{for } 9 2 \; U \text{”}$$

Figure 6: The percent-change next-price distribution with 129 bins based on historical Ethereum price data.

In practice, we solve this constrained optimization problem using the sequential least squares programming (SLSQP) method [9].

5 EXPERIMENTS ON HISTORICAL ETHEREUM PRICE DATA

5.1 Ethereum price data

To study the liquidity provision strategies described in the previous sections, we create a next-price distribution from historical Ethereum price data. We use the 10-minute ETH price from March 2018 to April 2020 (100,000 observations). We calculate the percent change in price at every time step, where

$$\text{percent change} = 100 \; {}^{1}\%_{e \, , \, 1} \quad \%_{e} {}^{o} \bullet \%_{e} \text{”} \tag{17}$$

Figure 6 shows the probability distribution of this percent-change next-price ETH distribution, for 129 bins on the interval 3% 3%¼ with each bin containing about 0”046% We use this distribution to govern the stochastic process of the price changes for simulating the return on di erent liquidity allocation strategies.

5.2 Proportional strategy (Risk neutral provider)

We rst study the expected utility of a risk-neutral liquidity provider (0 = 0) with a proportional allocation strategy for di erent choices of U and g, where utility is a function of the expected rewards at each time step given a providers risk preference.

Figure 7 (left) shows a density plot of expected utility. Figure 7 (right) shows slices of the surface, rst xing di erent choices of =U and then xing di erent choices of =g. We include two di erent kinds of calculations. The solid lines are based on the analytical results (plugging in the Ethereum next-price distribution to Equation 13). The symbols represent the sample-average utility coming from a direct simulation of the strategy for 50,000 time steps of simulated data. This con rms the validity of the analytical, Markov-chain based model.

For a risk-neutral liquidity provider, the highest expected utility occurs when there is just a single U and g bin, i.e., the current bin, 1B. In this strategy, the provider allocates all of their liquidity to the single, highest probability bin (which has a probability mass of

Figure 7: The expected utility of di erent proportional strategies for a risk-neutral provider (0 = 0). Left: a density plot of the analytic expected utilities, varying number of g bins and U bins. Right: Varying g holding number of U bins xed and varying U holding number of g bins xed. The solid lines denote the results of the analytical, Markov analysis (plugging in the ETH next-price distribution). The symbols denote the result of simulating the strategies directly on 50,000 time steps.

Figure 8: Optimal distribution for the liquidity allocation of a g-reset strategy with parameter =g set to the smallest value such that g contains at least 50% of the probability mass of the next-price distribution. The gure displays the density function of the optimal liquidity allocation. A risk-seeking (0 = 1) or risk-neutral (0 = 0) provider puts all liquidity into the current-price bin. As the provider becomes more risk averse (e.g., 0 = 0”1•1•10), it prefers to use more bins to reduce variance. An extremely risk-averse provider (e.g., 0 = 15) makes a uniform allocation over the U bins.

about0”15). Since the provider is allocating 100 units of liquidity, this corresponds to a utility of 15 per time step.

5.3 Optimal liquidity allocations

Figure 8 demonstrates the optimal liquidity allocation for a provider with di erent levels of risk aversion. This is for a g-reset strategy with parameter=g set to the smallest value such that g contains at least 50% of the probability mass of the next-price distribution. For a risk seeking0(= 1) or risk neutral (0 = 0) provider, the optimal allocation uses the single, current-price bin. This allocation results in a high expected reward, but a correspondingly high variance. As the provider becomes more risk averse (e0g=, 0”1•1), the optimal allocation makes use of more bins, while still putting a higher proportion of the liquidity into the higher probability, central bins. As the provider becomes very risk averse (e0g=, 10), the optimal allocation spreads to cover all bins inU, and for0 = 15the optimal allocation is uniform over these bins.

5.4 Comparing di erent strategies

Figure 9 compares performance of the optimal, proportional, and uniform g-reset strategies for di erent risk preferences. In each case, we de ne=g as the smallest value such that g contains at least 50% of the probability mass of the next-price distribution.

At risk-neutrality (0 = 0) and very low risk-aversion (e.g0., = 0”1) the proportional strategy is almost optimal with b = 0”14 and U = 0”74respectively. At high levels of risk aversion (e.g0.= 10), the uniform allocation is near-optimal and for an extremely risk averse provider (e.g.0 = 15), the optimal allocation is exactly uniform. This aligns with the optimal allocations presented in Figure 8.

Figure 10 demonstrates the optimal expected utility for various values ofg, whereg de nes the proportion of the probability mass of the next-price distribution that the set g contains. For the risk-neutral agent (0 = 0), they prefer small values ofg, because they are willing to update their allocation more frequently given the allocation is all the liquidity in a single bin. For a more risk-averse providers (e.g.0 = 3), they prefer a largeg and the resultingly larger number of bins to spread their liquidity over to reduce the variance in the rewards they receive.

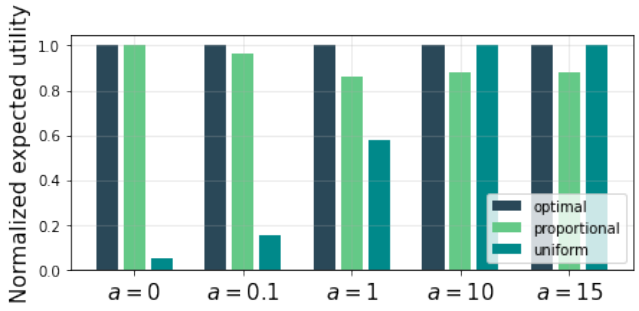

Figure 9: Expected utility of the optimal, the best proportional, and the uniform allocation for different risk preferences (values of a). For each level of a, utility is normalized by the optimal expected utility for that a. At lower levels of risk aversion (e.g., $a = 0, 0.1, 1$), the proportional strategy strongly outperforms the uniform allocation. At higher levels of risk aversion (e.g., $a = 10, 15$) the uniform strategy is optimal.

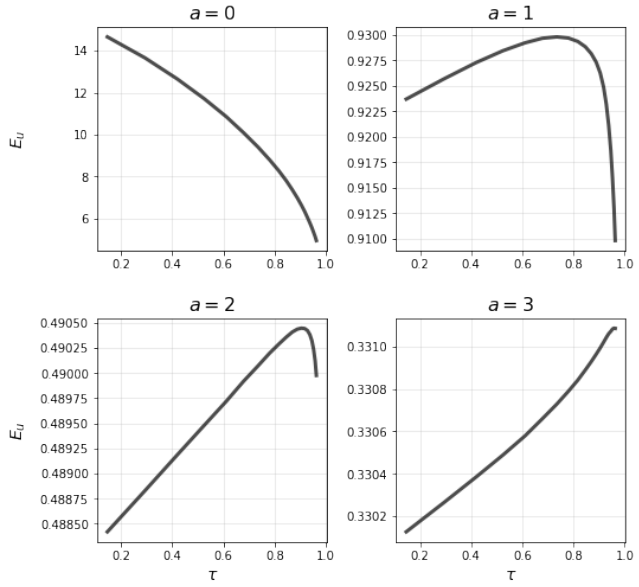

Figure 10: Expected utility from the optimal τ-reset strategy as a function of τ. Here, we report τ as the amount of probability mass that is contained in the B_τ bins. A risk-neutral provider ($a = 0$) prefers a small τ and is willing to update its allocation quickly. For a more risk-averse provider (e.g., $a = 3$), the opposite is true. Since they are spreading their liquidity over more bins, the optimal τ is as high as possible to cover more bins and further reduce the variance of their rewards.

5.5 Comparison with Uniswap v2

Given the historical price data of Ethereum, we can backtest the performance of strategic liquidity provision versus the Uniswap v2 liquidity provision. Recall that in Uniswap v2, liquidity providers aren't able to specify the price intervals over which they want to provide liquidity.

Figure 11 shows the historical ETH price for the time period that we use. As denoted in the inset, the black line represents the price of Ethereum at that time period, while the red and blue lines represent the bounds of the α and τ bins respectively for the optimal τ-reset strategy where $\tau = 0.5$ of the next-price distribution and $a = 0.1$ (top right allocation of Figure 8). This demonstrates how the strategy adjusts the liquidity provision regularly to recenter on the current price.

To simulate a Uniswap v2 liquidity provision strategy, we allocate uniformly over the entire price range of the data. In this example, the price ranges from 81 USD to 830 USD, and we discretize this range of prices into about 5500 bins, based on the spacing defined from the empirical calculation of the next-price distribution (Figure 6).

We allocate liquidity evenly among these 5500 bins, providing a guaranteed utility of $u(1/5500, a)$ at each time step. We compare this to v3 by finding the optimal τ-reset strategy and dividing the liquidity accordingly to calculate the utility of each price in the data set. The optimal $\tau-$reset liquidity provision strategy has $230x$ more utility than the v2 strategy for the slightly risk averse provider ($a = 0.1$).

6 CONCLUSION

This work explores the strategic liquidity provision problem that results from the Uniswap v3 protocol. We present the broad τ-reset class of strategies, and outline a technique for analytically calculating the expected utility of them. We describe three different implementations of the strategy, and compare their performance on the next-price distribution created from the historical Ethereum price data. We are able to find the optimal τ-reset strategy, given a choice of τ bins, and a next-price distribution. By backtesting our strategy on the historical price data, we find that the optimal τ-reset strategy can have an expected utility that is over 200x the utility a v2 strategy.

We hope that this work serves as a first step in formalizing and comparing the performance of these strategies. This present framework only represents a subset of the full strategy space, considering strategies that use the same approach to allocation across time, with the strategy centered on the current price and a reallocation triggered by a large enough price movement. A richer class of strategies would also modify the allocation of liquidity, and reset strategy, based on recent trends in price movement.

It will be interesting to investigate the question of liquidity provision in a multi-provider context. Whereas in this paper we study the investment problem facing a single v3 player vs a set of v2 players, we see an opportunity in future work to study emergent, equilibrium properties of this Uniswap v3 liquidity provision setting. An empirical study of strategies pursued on Uniswap v3 is also of interest. Additionally, there are intriguing macro-level connections between Uniswap v3 and gas prices. If gas fees are low, then liquidity providers are motivated to update their positions more