B.AMM - Efficient Automated Market Maker for DeFi liquidations

Dr. F Mr. C

June 2021

Abstract

Liquidations reside at the core of lending platforms, synthetic assets, derivatives and stable coins. The Backstop Automated Market Maker (B.AMM) is an automatic market maker optimized for lending platform liquidations. It is a fully autonomous smart contract and can efficiently handle liquidations of big debt with smaller capital requirements.

1 Introduction

Decentralized lending platform, e.g., Maker [9], Compound [11], dydx [10], bZx [6], and Aave [1], notoriously enable poor leverage ratio of x3-x5, despite having billions of dollars of liquidity at decentralized exchanges, that can facilitate liquidations at time of need. Lending platforms are being conservative as most of DeFi liquidity is concentrated on automated market makers (AMMs) who either (1) offer relatively high slippage w.r.t deposited amount (e.g., Uniswap V2 [13], Sushi, Balancer [3], and Bancor [4]); or (2) offer tight spread but can get depleted upon price changes (e.g., Uniswap V3 [14], Kyber's DMM [8]). Hence, AMMs like Uniswap V2, will fail to facilitate $20M DAI liquidation, despite having over $200M deposited inventory at its ETH/DAI pool.

A prominent AMM that supports both low slippage and infinite price range is Curve Finance, however so far it was limited only to support asset pairs with tightly correlated price, e.g., USDT and DAI. In this manuscript, we present the *Backstop AMM* (B.AMM) which extends, with the help of a price feed, Curve Finance's stable swap invariant into two arbitrary pair of assets (e.g., ETH and DAI). As a result the B.AMM prices pairs according to their current market price, but with potential discount or premium according to the imbalance of the position (namely, higher imbalance will result at a higher price discount), and the discount (premium) is calculated according to Curve Finance's stable swap invariant.

This gives rise to a fully automated and autonomous system that: 1. Pool user funds at a yield bearing platform, e.g., Uniswap V2 or YFI. 2. Upon demand, when liquidation is needed, withdraws some of the pooled funds (e.g.,

DAI), and executes the liquidation (e.g., give DAI in return to ETH with 8% discount). 3. Put the seized collateral for sale at the B.AMM, with a discount on market price, while maintaining a lower discount than the liquidation discount. 4. Upon every collateral sell, the return, e.g., DAI, is deposited back to the user funds pool.

We show that such system can handle liquidations much better than constant product AMM, given the same deposited funds, and can arguably give better yield to the users, subject so some additional risk.

In the next Section we describe the system. Then, in Section 3 we give the details on how to adjust Curve Finance's stable swap invariant into an arbitrary pair. In Section 4 we present a conservative simulation that shows the system can handle real world CeFi size liquidations.

2 System Design

At the proposed system, users provide liquidity that is used for liquidations (e.g., repay DAI debt in return to ETH collateral), and after liquidation happens, an automatic re-balance process begins. The re-balance process converts the seized collateral back to the original asset (e.g., the ETH collateral is converted back to DAI). The rebalance is done by offering the collateral for sale according to the market price, which is determined according to a price oracle. An optional discount on market price is given according to the imbalance size (the size of collateral to sell), and the exact formula is depicted in Section 3.

As user deposits are expected to sit idle for the majority of the time (when liquidations do not occur), the system will deposit it, on behalf of the users, to a yield bearing protocols, e.g., Uniswap or Compound, and will withdraw it only to facilitate liquidations.

Formally the system is composed of four main components, namely:

1. The **IDLE** component stores pooled liquidity (user deposits), and can use the deposited funds when they are not used for liquidations. E.g., an **IDLE** component can allow DAI deposits, and while the DAI is not used for liquidations, the DAI will be deposited in Compound. It can also allow multiple deposits, e.g., DAI and ETH, and put those funds at Uniswap V2.

2. The **BITE** component connects the **IDLE** component to lending platforms that rely on the B.AMM for liquidations. Upon request it withdraws funds from the **IDLE** component, executes a liquidation, and transfers the returned collateral to the **REBALANCE** component. E.g., it withdraws DAI from the pooled liquidity, and transfers the ETH that was received during the liquidation. If the pooled liquidity is, e.g., deposited in a DAI/ETH Uniswap pool, then both DAI and ETH will be withdrawn, but only DAI will be transferred.

3. The **REBALANCE** component is in charge of selling the liquidated collateral, in return to the underlying token of the liquidated asset. The

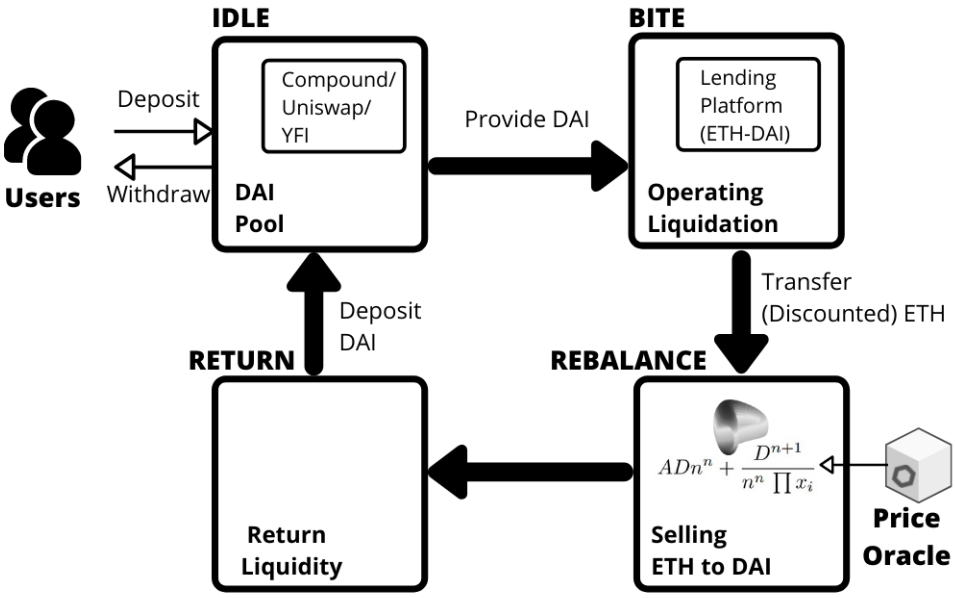

Figure 1: B.AMM high level design.

return of the sale is transferred to the **RETURN** component.

4. The **RETURN** component deposits the funds back in the **IDLE** component. This operation could be trivial, e.g., a simple deposit of DAI to compound, or more complex operation, e.g., return ETH/DAI liquidity back to a Uniswap pool. The latter is more complex, as it requires a logic to protect against Uniswap price manipulation.

The interaction between the four components is depicted at Figure 1 and described in details in the next subsection.

2.1 IDLE

The **IDLE** supports four functionalities, two of them are used by the protocol liquidity providers, one by the **BITE** component, and one by the **RETURN** component.

Deposit: Users pool funds by depositing to the **IDLE** component. Every user deposit issues a share in the pool according to the deposit relative size, w.r.t the inventory that already resides at the **IDLE**, **REBALANCE** and **RETURN**

components. If applicable, the price feed is used to normalize the value of the asset types at the **REBALANCE** module.
Withdraw: A user can withdraw her funds proportionally to the amount of shares she owns. Assets outside the **IDLE** component are withdrawn as is, proportionally to the amount of shares, without converting them back to the deposited assets. The withdraw process does not rely on a price feed.
Bite: In liquidation, a debt of type D with amount d is being repaid in return to a collateral of type C. Upon such call, the **IDLE** component takes d tokens of type D, from the pool liquidity and transfer it to the **BITE** component. In the general case, e.g., if the idle funds are deposited at Uniswap, in order to withdraw D token, it might be needed also to withdraw other assets. If this is the case then those assets are sent to the **RETURN** component.
Return: Some yield bearing platforms, e.g, Uniswap, requires for the deposits to have a basket of coins in certain ratio. Hence, when withdrawing a specific asset to serve the bite operation, it might be needed to withdraw other assets as well. E.g., when withdrawing DAI from an ETH/DAI Uniswap pool, one also have to withdraw the equivalent amount in ETH. Similarly, when depositing back to Uniswap, one have to deposit equal amounts of DAI and ETH. The return function takes care of this this logic, e.g., to deposit back only equal amounts of ETH and DAI.

2.2 BITE

The **BITE** component provides the liquidation interface for external lending platforms. Whenever an account in the lending platform is subject to liquidation, a *bite* transaction can be invoked by anyone. The call repays the liquidated account debt (e.g., DAI) by withdrawing the needed funds from the **IDLE** component, and sends the account's collateral (e.g., ETH) to the **REBALANCE** component, which in turn, will sell it back to the original asset (e.g., will sell the ETH in return to DAI).

The amount of collateral per debt unit is determined by the lending platform (typically it is 105-108%), however the **BITE** component may choose to reject a liquidation request, if, e.g., the lending platform price feed deviates from the B.AMM price feed. This allows a security model which is oblivious to any underlying bug at the lending platforms. In other words, the B.AMM can safely integrate with any lending platform without reviewing its underlying security.

2.3 REBALANCE

During the liquidation process, and inventory imbalance is created. Indeed, if, e.g., DAI debt is repaid in return to ETH collateral, the total portfolio of the system appreciate in value, thanks to the liquidation discount, however the portfolio now has excess ETH, and a shortage in DAI. The role of the **REBALANCE** component is to sell the excess ETH back to DAI. This is done by offering the imbalance for sale according to a special price function, which we describe at Section 3.

Given the price function, a smart contract logic allows anyone to buy the ETH imbalance, in return to DAI, according to the price function. Where the price function depends on the imbalance and the market price (as reflected by the price feed oracle). The bigger the imbalance, the higher the discount over market price will be, and at imbalance that is close to 0, the price will be identical to the market price.

The reader should note that even though the sell is done below market price, a profit is expected, as the sell discount will always be smaller than the liquidation discount. Nevertheless, a loss could occur, if there are sharp market movements before the sell is completed. Our simulation at Section 4 demonstrates that despite intermediate losses, this operation is expected to be mostly profitable at the long run.

2.4 RETURN

The return of the sold imbalance should be deposited back to the **IDLE** component. A dedicated logic can be implemented by the **RETURN** component in two cases:

- When the **IDLE** deposit operation is gas consuming, then the component can make sure only large amounts are deposited. This saves the gas cost overhead, which could otherwise potentially consume most of the trading profits.
- When the **IDLE** deposit requires a special logic. E.g., consider the case where the **IDLE** component deposits user funds into an ETH-DAI Uniswap V2 pool. In such case, upon liquidation both ETH and DAI will have to be withdrawn, while only DAI will be used in the liquidation. Whenever the excess ETH is sold back to DAI, by the **REBALANCE** component, the **RETURN** component will deposit equal amounts of DAI and ETH back to the Uniswap pool.

In the case of a blockchain with low gas costs (e.g., and L2) and if the user funds are deposited in a simple yield bearing product, e.g., Compound Finance, then this component can be degenerated, and simply deposit any accumulated funds back to the pool.

3 The Stable Swap Invariant

In this Section, we present a novel automated algorithm (pricing formula) for rebalancing a portfolio of assets, namely a formula that decides on a sell price of an asset, given the current portfolio size and imbalance, and given current market price (which is taken from an oracle price feed).

The DeFi ecosystem has a long line of work of developing automated pricing formulae, e.g., Uniswap [13], Bancor [4], Balancer [3], DODO [7], Kyber Network [8]. However all of these previous works were focused on *market making*

where at every point of time the portfolio is assumed to be fully balanced. To the best of our knowledge, this is the first time an automated rebalance algorithm is investigated. At all previous work the *market making* algorithm assumes the portfolio is balanced, and its main functionality is to price a slippage for a given portfolio size and sell qty. In this work, the algorithm has full knowledge on the size of the imbalance (thanks to a price feed), and has to price a slippage for given portfolio size, imbalance size, and sell qty, where only excess inventory is being offered for sale. A rebalance algorithm differs from market making algorithm in two fundamental proprieties:

- In rebalance algorithm, bigger sell quantities result in lower price change (per qty unit). This is because the outcome of a bigger sell is a more balanced position. While in market making algorithms, when the sell quantity is bigger, the change in price (per qty unit) is bigger.
- In market making algorithm, the formula always tries to sell in a price that is more expensive than the estimated market price. This is to compensate the risk of incurring into an imbalanced situation after the sell. In rebalance algorithm, the ask price would be usually lower than market price. Note that in our case the imbalance increases only upon liquidations, which are done with significant discount over market price. Hence, rebalancing below market price could still be profitable.

Curve Finance AMM. A notable exception to the AMMs we surveyed above is Curve [12]. Curve Finance AMM is using the stable swap invariant to price assets. It's core property is that it has a target portfolio (i.e., target ratio between two or more assets) and it provides different slippage according to the distance from the target portfolio, and the bigger the target is, the bigger the slippage is.

This property makes it ideal to use the stable swap invariant for the automated rebalancing process. However, the stable swap invariant was tailored to correlated asset classes, e.g., DAI and USDT. As our system already relies on an external price feed for the liquidation process. Hence, we can normalize all portfolio assets to their USD values, and plug it into the stable swap invariant, without additional security risk.

The stable swap invariant gives a price for infinitesimals small qty, provided the inventory of two assets X, and Y. Formally, the stable swap invariant (for two assets) is:

$$A(X+Y)+D = 4AD + \frac{D^3}{4XY}$$

where

$$D = X + Y$$

and A is a chosen amplification factor. The higher A is, the lower the slippage is.

The main property of the invariant is that the slippage is smaller when $\frac{X}{Y}$ is closer to 1 (i.e., when $X = Y$). This property is very desirable for a

rebalance algorithm, hence we chose to use the invariant as is, with the following adjustments:

- We normalize all inventory to USD value (using a price feed).
- We denote by inv the amount of inventory we have over all 4 components. And by imb the amount of inventory we have only at the **REBALANCE** component. Then we set $X = inv - imb$, and $Y = inv + imb$, and the asset for sale is of type Y.

This design choice guarantees that Y is priced with discount over market price as long as $imb > 0$, and the bigger imb is, the bigger the discount is. Further, Curve Finance implemented a $get\_x(X, Y, dy)$ function that returns the amount of X assets that are given in return to dy quantity. Hence, B.AMM formula, is

$$get\_x(inv - imb, inv + imb, dy)$$

if $imb > 0$ and 0 otherwise.
Choice of A. The amplification coefficient A should reflect the risk appetite of the AMM. Higher A will offer lower discount on market price, which will lead to bigger imbalance, but also higher expected profit. A line of research for future work is to design a dynamic A that depends on market volatility.

4 Simulation over CeFi Liquidations

In this Section we simulate the B.AMM w.r.t real world, CeFi liquidations from exchanges that allow x100 leverage. We first develop a modeling framework to such simulation, and then perform the simulation over Binance Futures real liquidation data. Both our theoretical modeling and the simulation shows the B.AMM approach is up to x20 times more capital efficient w.r.t existing AMMs, and can handle over 1 billion USD of monthly liquidations.

4.1 Theoretical model

In order to simulate the system, we need to construct a model that describes how the **REBALANCE** model will perform. Namely, how quickly it will rebalance the inventory, and what will be its trading profit and loss (PnL). We follow Gauntlet [2] line of work, and build a conservative simplified model for the **REBALANCE** model. In our model, the B.AMM has inventory of size inv_b, and there is only one additional source of liquidity in DeFi, namely **DEX**, which adheres to Uniswap constant product formula, with inventory inv_d for each of the AMM assets (e.g., if it has \$100M of USDT and \$100M of ETH, then $inv_d = 100M$). We further assume that whenever there is an arbitrage between the **REBALANCE** price and the **DEX** price, it will be taken immediately. Finally, we assume a time interval T, and allow only one arbitrage to be done (between **REBALANCE** and **DEX**) in that time interval. Intuitively, T represents the time duration that will take for DeFi liquidity to

recover, after every big trade. E.g., if $inv_d = 100M$, and $T = 30$, and an arbitrage of $3M$ USD was done, then after 30 minutes the **DEX** will return to $inv_d = 100M$, and the price will be according to the real market price (taken from CeFi exchanges).

This approach is conservative as in practice not all of DeFi liquidity is concentrated at constant product exchanges, and in practice even before the time interval ends, there will be some kind of partial rebalance in the **DEX** position.

Given the above model, we measure two metrics:

- The percentage of missed liquidations, namely, the amount of volume the system was not able to liquidate. And the conservative assumption is that if at the exact moment of liquidation, the system did not have sufficient fund to repay the liquidated debt, then the entire liquidation was missed.
- The PnL of the system in a given time frame.

The goal is to minimize the first parameter and to maximize the second parameter.

4.2 Experimental results

We performed a simulation according to Binance Futures ETH-USD long liquidations in the dates of 2021-04-10 to 2021-04-30 [5]. During those 20 days, over \$700M long [1] liquidations occurred, and as depicted at Figure 2 the liquidation distribution over time is far from uniform. We simulated the B.AMM perfor-

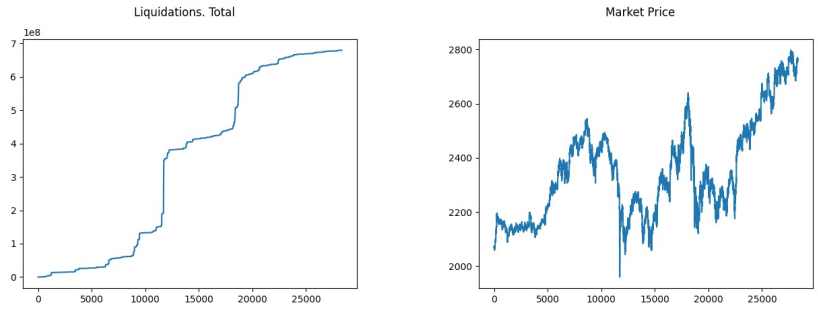

Figure 2: On the left hand side: ETH-USD long liquidations on Binance Futures, between the dates of 2021-04-10 and 2021-04-30. The X axis shows time in 30 minute units, and the Y axis shows total liquidations since the beginning of the period in units of \$100M. On the right: ETH-USD price.

mance for multiple A, T and inv_b values. As expected higher A values increase the percentage of missed liquidations, but also increases the PnL. However, as

[1] Taking also short liquidations into account would only improve our results, as it would make the rebalance process easier.

the PnL still remains quite high, we only present the results for a specific value, namely $A = 20$.

We first check the percentage of missed liquidations for different T and inv_b values. A sum of \$700M in 20 days amounts to \$1B monthly liquidations, and as depicted in Figure 3, the B.AMM outperforms constant product AMM. Under the most optimistic assumption, i.e., that DeFi liquidity is restored every 1 minute, the Uniswap AMM still misses over 20% of total liquidations, while the B.AMM is missing under 7% when using the same inventory size, and even 0% when increasing the inventory size. For the most conservative T value, Uniswap will miss almost half of the liquidations, while the B.AMM will miss only 2% if an inventory of \$200M is provided.

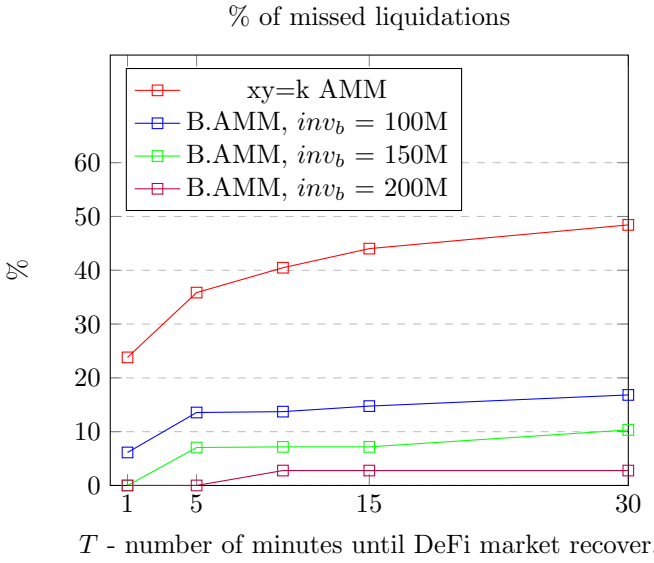

Figure 3: % of missed liquidations over Binance Futures ETH-USD long. For inv_d = \$100,000,000, $A = 20$, and liquidation penalty of 5%.

Bigger inventory gives rise to smaller miss percentage. Hence, the scalability of the B.AMM crucially relies on the willingness of active participants to deposit more funds. For this purpose we simulate the annual profit, based on the 20 days simulation, and for simplicity we assume that the **IDLE** component brings zero yield. The results are depicted in Figure 4, and suggest that to backstop \$1B monthly liquidations, even a capital of \$1B will bring decent yield. On contrast, Uniswap liquidity providers will not profit nearly as much from backstopping.

5 Conclusion

In this manuscript, we presented the B.AMM, a novel automated system to efficiently handle liquidations. We demonstrated that the system is superior

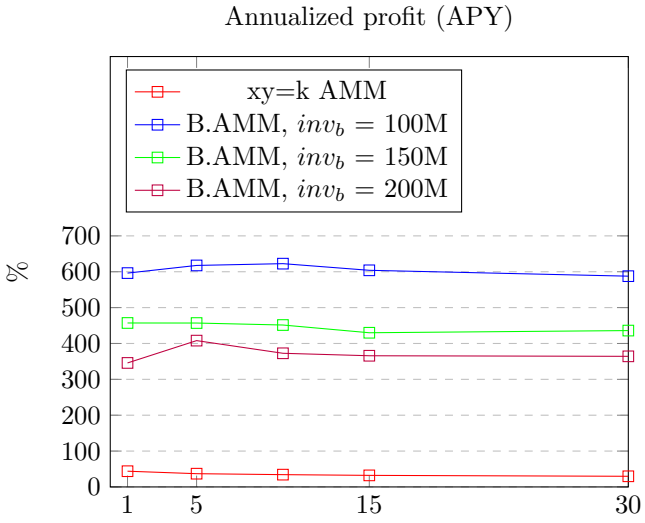

Figure 4: APY for inv_d = \$100,000,000, $A = 20$, and liquidation penalty of 5%.

over constant product AMMs, and can handle over \$1B of monthly liquidations. The system mitigates three of the most important pain points in DeFi lending platforms:

- Current lending platforms offer very poor leverage, due to lack of confidence in their liquidation process. With B.AMM x20 leverage can be easily supported.
- Lending platforms cannot support the long tail of assets, despite having massive amount of liquidity at Uniswap and Sushiswap. Encouraging the community to deposit their liquidity at the B.AMM, which in turn will route the liquidity to Uniswap or Sushiswap will give rise to guaranteed liquidatiry in times of liquidations. And lending platforms would be able to support these assets.
- Lending platforms today either rely on unknown keepers to liquidate at time of need, and some even have such keepers on their payroll. This setting cannot exist in a fully decenteralized and transparent system. The B.AMM give rise to a fully decentralized and transparent community based keeper system.

Liquidation resides at the core of lending platforms, synthetic assets, derivatives and stable coins. The B.AMM offer more efficient and decentralized solution for these platforms.

References

[1] *AAVE Protocol Whitepaper*. URL: https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf. (accessed: 9.06.2021).

[2] *An Analysis of the Market Risk to Participants in the Compound Protocol*. URL: https://scfab.github.io/2020/FAB2020_p5.pdf. (accessed: 9.06.2021).

[3] *Balancer: A non-custodial portfolio manager, liquidity provider, and price sensor*. URL: https://balancer.fi/whitepaper.pdf. (accessed: 9.06.2021).

[4] *Bancor Protocol*. URL: https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en.pdf. (accessed: 9.06.2021).

[5] *Binance Futures ETH-USDT liquidations 2021-04-10 to 2021-04-30*. URL: IPFS%20QmcoDfZh5K64CyMCBxrgVmCGkCPP6RwzB7DTnWeWemiAPo. (accessed: 9.06.2021).

[6] *bZx Litepaper*. URL: https://bzx.network/pdfs/bZx_lite_paper.pdf. (accessed: 9.06.2021).

[7] *DODO: A Next-Generation On-Chain Liquidity Provider Powered by Proactive Market Maker Algorithm*. URL: https://dodoex.github.io/docs/docs/whitepaper/. (accessed: 9.06.2021).

[8] *Dynamic Automated Market Making*. URL: https://files.kyber.network/DMM-Feb21.pdf. (accessed: 9.06.2021).

[9] *Introduction to the Maker Protocol*. URL: https://docs.makerdao.com/. (accessed: 17.03.2020).

[10] Antonio Juliano. *dYdX: A Standard for Decentralized Margin Trading and Derivatives*. URL: https://whitepaper.dydx.exchange/. (accessed: 21.03.2020).

[11] Geoffrey Hayes Robert Leshner. *Compound: The Money Market Protocol*. URL: https://compound.finance/documents/Compound.Whitepaper.pdf. (accessed: 02.03.2020).

[12] *StableSwap - efficient mechanism for Stablecoin liquidity*. URL: https://curve.fi/files/stableswap-paper.pdf. (accessed: 9.06.2021).

[13] *Uniswap v2 Core*. URL: https://uniswap.org/whitepaper.pdf. (accessed: 9.06.2021).

[14] *Uniswap v3 Core*. URL: https://uniswap.org/whitepaper-v3.pdf. (accessed: 9.06.2021).