Samuel Ballan
08/23/2021

Stability, Liquidity, and Governance

Decentralized Finance (DeFi) is a rapidly growing field in cryptocurrency. Two important services provided in DeFi are stablecoins and decentralized exchanges (dexes), and I will be discussing two of the most successful ones in this paper. Dai, a stablecoin, and Uniswap, a dex and automated market maker, have each been tremendously successful, as can be attested to by the many billions of dollars locked up in their protocols [4]. The Dai uses collateralized loans to maintain its dollar peg [5], and Uniswap uses user-provided liquidity pools to facilitate swaps between ERC-20 tokens [1]. Each makes use of arbitrage incentives in interestingly similar and different ways. After a discussion of the implementations and design goals of Dai and Uniswap, I will propose a new governance, investment, voting, and exchange protocol that I believe can fill a niche in the DeFi market.

Dai

Dai is specifically designed to be a token with very little price volatility that tracks the value of the US dollar [5]. Dai is generated when users of the Maker protocol deposit collateral in Maker vaults and take out loans of Dai [5]. Therefore, every token of Dai is backed by collateral that exceeds the value of a dollar[5]. When the market price of Dai falls below a dollar, users are incentivised to buy Dai at the lower price to pay back their debts and receive both their original collateral and profit in Dai. If the price rises too high, users are incentivised to take a loan of Dai, to be paid back when the price drops [5]. Like most decentralized stablecoins, Dai uses an external oracle to determine the value of the collateral and peg [6].

If the value of the collateral falls too much, it is liquidated and sold in a collateral auction. If the auction doesn’t raise enough money to cover the borrowed Dai, the deficit is converted to debt which is shared by the entire protocol, as opposed to a particular vault. There are a number of mechanisms used to handle this protocol debt, including the minting of the MKR token, sold in exchange for Dai. The MKR token is bought back and burned when there is a surplus of Dai [5]. The MKR tokens carry intrinsic value in their utility in governance of the Maker protocol [5,6].

Uniswap

The Uniswap protocol is designed to incentivise behavior that balances the tokens in their liquidity pools in quantities that reflect their market exchange rate [2]. To determine the exchange rate between tokens at a given time, Uniswap uses a constant function[1,2]. In Uniswap v2, a constant product function of $\boldsymbol{x * y = k}$ was used, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are each quantities of the two tokens in the pool, and k is a constant [2]. The spot price at any given moment is said to be $\boldsymbol{x / y}$, but the quantities being bought or sold will cause the price to slip. The price is set such that, once the exchange has been completed, the value of $\boldsymbol{k}$ remains unchanged. As can be seen in the following diagram from Aigner and Dhaliwal (2021), as the quantity of $\boldsymbol{x}$ tokens in the pool decreases, the price of $\boldsymbol{x}$ approaches infinity [3].

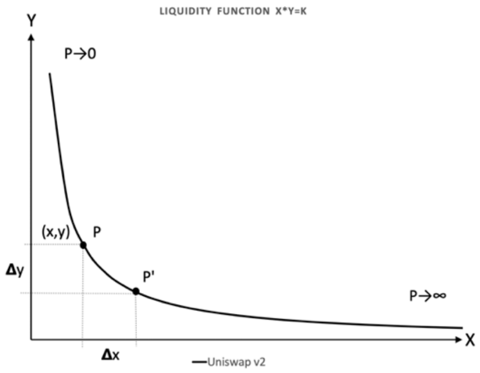
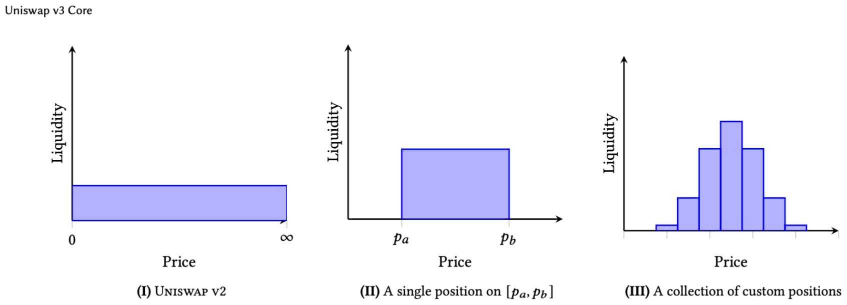


Figure 1: Liquidity function of Uniswap v2.

As the price moves further from the market price, the pool suffers greater amounts of impermanent loss. The pool will be forced to offer tokens at non-market prices, which arbitrageurs will quickly take advantage of. The total amount of value in the pool will decrease, compared to what it would have been if no trades had happened at all [3].

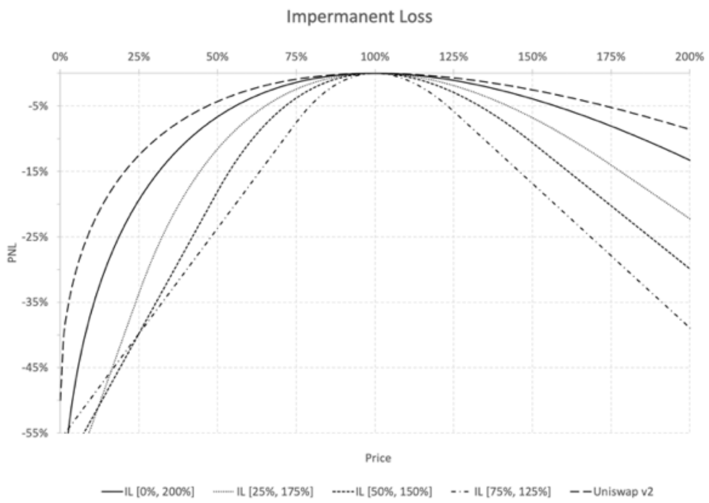
Uniswap v3 took significant steps to reduce the amount of impermanent loss [3]. First and foremost, instead of a single liquidity pool, the price range for the tokens is divided into liquidity positions [2]. Each position can be thought of as its own bounded pool, which only provides liquidity in limited price ranges. The quantity of tokens that can be swapped at a given price is increased by having many liquidity positions, and liquidity providers are incentivised to close positions that fall outside the current market price [2]. This means that liquidity providers that would have suffered great impermanent losses have their tokens untouched, and are encouraged to close their current position and open one closer to the market price [2]. In the following diagram from Aigner and Dhaliwal (2021), we see how a divided liquidity pool can provide maximal liquidity at a strategic point, instead of uniform liquidity across the entire price range [3].



(I) Uniswap v2 **(II)** A single position on $[p_a, p_b]$ **(III)** A collection of custom positions

Figure 3: Example Liquidity Distributions

The implementation of these divided pools is rather complex, and requires the price movements of the tokens to be calculated across each possible liquidity position, which amounts to doing a calculation for every basis point of movement [3]. Even so, there is no way to completely avoid impermanent loss, which is the reason Uniswap shares its transaction fees with its liquidity providers [1,2]. As can be seen in the following diagram from Aigner and Dhaliwal (2021), even a tightly bounded liquidity position can suffer significant losses in the worst cases [3].



While they provide very different services, both Dai and Uniswap rely on the behavior of external actors to perform the arbitrage required to achieve a desired price. The Dai protocol is designed to incentivise behavior that drives the price of the Dai coin towards one dollar. The Uniswap protocol, on the other hand, is designed to incentivise behavior that creates liquid markets at market prices. Dai will track the value of the dollar, while Uniswap tracks the value of traded pairs. Each requires users to deposit tokens in the protocol, either to be used either as collateral, or in a liquidity pool. Both are susceptible to loss of funds in the case of drastic price movements in the market value of the deposited assets, either through liquidation or impermanent loss. And lastly - as a seeming byproduct of these services - both have governance tokens with multi-billion dollar market caps [4].

GIVE

I propose a new Governance, Investment, Voting, and Exchange protocol that I'm calling GIVE. GIVE turns the incentives of protocols like Dai and Uniswap upside down. Instead of providing a service that requires governance, it provides governance-as-a-service, in the form of user-defined votes and protocol governance. Instead of the treasury being used to secure the protocol, the growth of the treasury is a primary goal of the protocol. At the same time, it does not carry the same risks of user loss as Dai or Uniswap. In fact, the primary risk is that the token will maintain its peg *too closely*.

Like Uniswap, users of GIVE can define their own liquidity pools. However, instead of a pool being merely a reserve of an arbitrary pair of coins, these pools contain a reserve of an underlying asset (eg, BTC) and a corresponding GIVE token (eg, gBTC), and a treasury of the underlying asset (eg, BTC). A GIVE token (generically, "gGIVE") is minted by depositing its corresponding asset in the liquidity pool's treasury. The treasury always maintains a ratio of at least 1:1, asset to gGIVE, ensuring that gGIVE can always be redeemed for the underlying asset. For example, a user could deposit 2 BTC to create a liquidity pool with a reserve containing 1 BTC and 1 gBTC, and a treasury containing 1 BTC. This would be an expensive, but simple, method of kickstarting a pool.

An asset's gGIVE counterpart is said to be an "Eventually-Stable-Coin", since it never drifts too far from its peg, but is also resistant to having its price quickly brought back to its peg. This resistance prolongs opportunities for arbitrage, which adds value to the pool's treasury in the form of transaction fees. There are two primary factors that make such resistance possible: 1) the liquidity pools can mint and burn gGIVE tokens as needed, and 2) users can always redeem underlying assets at the proper peg price, so long as they are willing to wait the duration of a lockup period. I will explain the mechanics that make this possible in a later section.

The treasury will take advantage of opportunities to reduce the ratio of gGIVE to asset tokens, resulting in a surplus. This may happen if the treasury is able to buy back and burn gGIVE at below the peg, or gGIVE is minted for a price above the peg. Unlike Dai, gGIVE tokens are backed by collateral that perfectly matches their peg, so there is no need for over-collateralization. The treasury's surplus is invested, and the returns on this investment become the subject of the gGIVE token's governance votes. Holders of gGIVE can decide where this money should go. It could be used to pay dividends to holders of the gGIVE, but it is unlikely that competitive rates could be provided. Instead, the intention of the GIVE protocol is for users to decide on ways to pool this money: open-source code developers they want to hire, artists they want to commission works of art and NFTs from, or causes that they wish to donate to. In other words, the intention is for users to pool the dividends together, toward a common purpose.

Finally, each time the investments (accrued by imbalances in the liquidity pools' treasuries) compound, a protocol token (pGIVE) is mined, in proportion to the amount that has been compounded, and the market price of the underlying asset. The pGIVE tokens are given to stakers of the gGIVE tokens, in proportion to the amount staked. The intention is to mint an amount of pGIVE that is proportional to the amount of growth the investment has seen. The more that it continuously grows, the more pGIVE tokens go into circulation. These tokens are used to govern the underlying GIVE protocol itself, much like the UNI and MKR tokens. As the size of the investments increase, the rate of growth may decrease, meaning that fewer pGIVE coins are minted.

Eventual-Stability: Slow Volatility

GIVE liquidity pools have the advantage of being native to the protocol, so they have the ability to mint and burn gGIVE tokens at the peg price. Additionally, user contributed assets can be used to provide liquidity, without the risk of impermanent loss. This allows native gGIVE pools to provide more competitive prices than any external liquidity pool could. Because external liquidity pools, bots, and users can all trade with native GIVE liquidity pools, there is incentive for GIVE liquidity pools to represent the true market price of any gGIVE token, which is reflected in the ratio of gGIVE to asset tokens in a given pool. All of these factors together make it possible to artificially control the price movements of a gGIVE token relative to its underlying asset. GIVE will allow the peg to move widely, but will ensure that it moves slowly. The length of the lockup period for minting and redemption and the price at which the liquidity pools will burn and mint coins can be all tuned to control the speed at which the peg will move. Further research will be required to determine exactly how wide the peg could be, but I estimate that the peg could be allowed to drift from 75% to 150% of the underlying asset.

gGIVE Below Peg: Buyers

When a gGIVE token is below its peg, the liquidity pool has more gGIVE than it has the underlying asset. Users will be able to take advantage of arbitrage opportunities with a few different strategies: Buy-and-Wait, Buy-and-Deposit, or Staking.

The Buy-and-Wait strategy is straightforward: users can buy the coin at a price below its peg, and wait for the price to come back up. It doesn't matter whether the coins are purchased from a GIVE liquidity pool or from an external exchange. This strategy requires the user to have confidence that the price will indeed come up again. If the user wants a guarantee of a higher price, they can use the Buy-and-Deposit strategy, where after purchasing the coin, they lock it up and receive the full asset value after a predetermined lockup period.

Given the guarantee of a higher price with Buy-and-Deposit, users are incentivized to buy up the gGIVE in the liquidity pool (and any external pools) until the price moves close to the peg, and then deposit all of it in lockup. With the peg nearly restored, there will be nearly equal amounts of gGIVE and the asset token in the liquidity pool.

There are two problems with these strategies. The first is that such arbitrage opportunities will stabilize the price of the coin very quickly, so opportunities for arbitrage will be limited. The second is that fast actors will always win, which means that bots will perform most of the arbitrage.

This brings us to the Staking strategy, which I also call “Rent-a-Bot”. A user can stake the underlying asset token, and be given priority access to the arbitrage opportunities. When there is too much gGIVE in the pool and the price falls too far below the peg, the stake will be used to immediately purchase some gGIVE at that low price, and offer the coin for sale to the pool at a slightly higher price. This mimics the behavior of an arbitrage bot, but it has access to prices that no external bot ever could have. When a user or external bot tries to buy from the liquidity pool at this low price, price slippage in the pool can be limited by buying some of the coins from the staker at that slightly higher price, with profits on the sale split three ways between the staker, the reserve, and the treasury. In this way, the staked asset acts as “backup liquidity providers”, without the risk of impermanent loss. Staked assets can therefore be truly thought of as investments, with the transaction fees paying a dividend.

gGIVE Below Peg: Sellers

While users are incentivised to buy gGIVE tokens when the price falls, some may still want to sell. It will be difficult for our liquidity pool to buy gGIVE when it doesn’t have much of the underlying asset token left, so sales of gGIVE would drive the price down even further. To ensure we can maintain the loose peg, staked assets can help us. Staked assets will be used to buy gGIVE on behalf of the liquidity pool to reduce price slippage. The staker can then redeem the gGIVE for a higher price from the treasury.

gGIVE Above Peg: Sellers

When a gGIVE token is above its peg, the liquidity pool has less gGIVE than it has the underlying asset. Users will be able to take advantage of arbitrage opportunities in a few different ways: Borrow-and-Wait, Mine-and-Sell, or Staking.

With the Borrow-and-Wait strategy, users can effectively short gGIVE by borrowing it at a high price and selling it, waiting for the price to come down, and then buying it again to pay back their debt. Users must have confidence that the price will come down, and must have access to collateral for the loan.

With the Mine-and-Sell strategy, users can wait until the market price of gGIVE is above the mining price, they can mine it, and then sell it. The price of mining generally changes with the price gGIVE in the liquidity pool such that this is only profitable when the price of gGIVE is exceptionally high.

With the Staking strategy, stakers get special access to a slightly lower mining price. The staker's asset tokens will be used in a Mine-and-Sell strategy with a reduced mining price. Although the mining price is reduced, it must still be at or above the peg.

gGIVE Above Peg: Buyers

While users are incentivised to sell gGIVE when the price rises, some may still want to buy. It may be difficult for our liquidity pool to sell gGIVE when it doesn't have many gGIVE tokens left, so buys may drive the price up even higher. To ensure we can maintain the loose peg, Staking can help us. Stakers will mine the coin at a lower price than the current market price (but above the peg), and then sell it to the liquidity pool at market price, ensuring that the price doesn't slip too high.

Use Cases

Users can get started with the GIVE protocol by staking an asset in one of the liquidity pools. Over time, and assuming price volatility of the gGIVE token they have stake with, this will earn them a slightly greater number of gGIVE tokens than the number of asset tokens they deposited. With a slowly growing collection of gGIVE tokens, a user can begin to use their tokens to vote. At the end of each voting cycle, returns from the GIVE investments are sent to wallets that were voted on by gGIVE voters. Users with more gGIVE will have more votes, and therefore can control a greater fraction of the donated returns. After voting, gGIVE tokens used for voting are returned to all users. A user may then either decide to participate in future votes, sell the gGIVE at market price, or lock up the gGIVE to receive the corresponding asset at the pegged price.

In essence, users can use gGIVE to make their voices heard in the world, without directly spending any money, and while maintaining exposure to a crypto asset that they like. If I like holding BTC, I can trade it in for gBTC, and start voting, all the while knowing that I can get back my original BTC in the future. If I want to support the protocol, and make some money, I can invest my original assets by staking in a liquidity pool. The GIVE protocol allows users to express their values using their money, but without spending it.

Future Research

Once there is more than a single liquidity pool in the GIVE protocol, there is the opportunity to create Uniswap-style liquidity pools: instead of pairs of asset tokens and corresponding GIVE tokens, they could be pairs of different GIVE tokens (eg, gBTC and gETH). Like Uniswap, the exchange rate of the tokens would be set by user arbitrage, so no outside Oracle would be required. However, by making use of the underlying base-layer GIVE liquidity pools, these pools should be resistant to price slippage and impermanent loss. The base-pool could burn a gGIVE token, and then swap an underlying asset token on the market, and use that token to mint a new gGIVE token, which it can give to the pool. Unfortunately, this would reduce the liquidity of the base liquidity pool, but the advantages of inter-gGIVE liquidity make it a worthy topic of further research.

Final Thoughts

DeFi is still in its infancy, and sometimes it feels like new protocols are released every day. While there is tremendous financial opportunity, I believe there is also an opportunity for collaborative work. The very core of blockchain technology is the ability for decentralized computers to arrive at consensus, and I believe that this also provides a platform for humans to arrive at consensus. I intend to continue research and development on this project, and hope that it may create opportunities for people to work together towards common goals.

References

1. Adams, H., Zinsmeister, N., & Robinson, D. (2020, March). *Uniswap v2 Core*. UNISWAP PROTOCOL. https://uniswap.org/whitepaper.pdf
2. Adams, H., Zinsmeister, N., Salem, M., Keefer, R., & Robinson, D. (2021, March). *Uniswap v3 Core*. UNISWAP PROTOCOL. https://uniswap.org/whitepaper-v3.pdf
3. Aigner, A., & Dhaliwal, G. (2021). UNISWAP: Impermanent Loss and Risk Profile of a Liquidity Provider. *SSRN Electronic Journal*. Published. https://doi.org/10.2139/ssrn.3872531
4. CoinMarketCap. (n.d.). *Cryptocurrency Prices, Charts And Market Capitalizations*. Retrieved August 24, 2021, from https://coinmarketcap.com/
5. *The Maker Protocol White Paper*. (2020, February). MakerDAO | An Unbiased Global Financial System. https://makerdao.com/en/whitepaper/
6. Moin, A. (2019, September 18). *A Classification Framework for Stablecoin Designs*. ArXiv.Org. https://arxiv.org/abs/1910.10098