

# A SOLUTION ON OPTIMAL ONE SIDED LIQUIDITY MINING IN DEX

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE AWARD OF THE DEGREE  
OF

**MASTER OF TECHNOLOGY  
IN  
SOFTWARE ENGINEERING**

Submitted by

**ANSHUMAN CHATTERJEE**  
**2K21/SWE/05**

Under the supervision of

**Prof. RUCHIKA MALHOTRA**  
(Professor and Head of Department)



DEPARTMENT OF SOFTWARE ENGINEERING  
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)  
Bawana Road, Delhi 110042

MAY 2023

M.Tech (Software Engineering)

Anshuman Chatterjee

2023

**DEPARTMENT OF SOFTWARE ENGINEERING**  
**DELHI TECHNOLOGICAL UNIVERSITY**  
(Formerly Delhi College of Engineering)  
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Anshuman Chatterjee, Roll No – 2K21/SWE/05 student of M.Tech (Software Engineering), hereby declare that the Project Dissertation titled “A Solution on Optimal One Sided Liquidity Mining in DeX” which is submitted by me to the Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Date: 30/05/2023

*Anshuman Chatterjee.*  
30/5/2023

Anshuman Chatterjee  
2K21/SWE/05

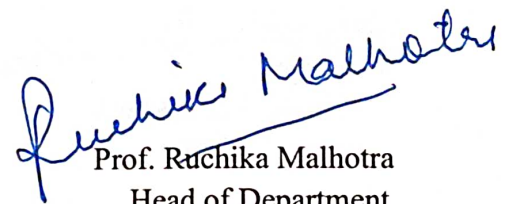
**DEPARTMENT OF SOFTWARE ENGINEERING**  
**DELHI TECHNOLOGICAL UNIVERSITY**  
(Formerly Delhi College of Engineering)  
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled “A Solution on Optimal One Sided Liquidity Mining in DeX” which is submitted by Anshuman Chatterjee, Roll No – 2K21/SWE/05, Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date: 30/05/2023



Prof. Ruchika Malhotra  
Head of Department  
Department of Software Engineering

**DEPARTMENT OF SOFTWARE ENGINEERING  
DELHI TECHNOLOGICAL UNIVERSITY  
(Formerly Delhi College of Engineering)  
Bawana Road, Delhi-110042**

**ACKNOWLEDGEMENT**

I wish to express my sincerest gratitude to Dr. Ruchika Malhotra for her continuous guidance and mentorship that she provided me during the project. She showed me the path to achieve my targets by explaining all the tasks to be done and explained to me the importance of this project as well as its industrial relevance. She was always ready to help me and clear my doubts regarding any hurdles in this project. Without her constant support and motivation, this project would not have been successful.

Place: Delhi  
Date: 30/05/2023

*Anshuman Chatterjee.*  
30/5/2023

Anshuman Chatterjee  
2K21/SWE/05

## ABSTRACT

Decentralised crypto exchanges or DeX has created a revolution in the Decentralised finance industry. With the creation of Uniswap and other defi lending-borrowing protocols, the DeFi industry has seen a meteoric growth in the late 2020. Currently the market capitalization of Defi industry is around 92 billion dollars in which the Dex is contributing around 60-70% of the total market capitalizations. DeX offers a lots of services like swapping of tokens, creating liquidity pools, lending and borrowing, flash loans, staking and harvesting. These services are controlled by Smart Contracts which contains set of instructions that executes itself automatically when some conditions are met. The rate of swapping of tokens is decided by a mathematical formula which usually called as AMM or Automated Market Makers. The AMM decides the rate or price of a token depending upon the amount of token pairs that are available in the liquidity pool. Every token swap that happens on the DeX is accompanied by a liquidity provider fee. This liquidity provider fee is distributed among the address that have provided liquidity to that pool. Liquidity is always added in token pairs. The AMM also decides the amount of token pairs that we need to provide for addition of liquidity into a already created pool. In this paper we are going to provide a mathematical solution and smart contract function for addition of liquidity in a pool using a single token. The solution will be highly optimized and will be tested in the popular DeX Uniswap.

**Keywords** : DeX, Liquidity Pools, Smart Contracts, AMM, Uniswap, Cryptocurrencies, Swapping, Liquidity Mining, Constant product market maker, Token pairs.

# CONTENTS

|  |     |
|--|-----|
| <b>Candidate's Declaration</b>               | i   |
| <b>Certificate</b>                           | ii  |
| <b>Acknowledgement</b>                       | iii |
| <b>Abstract</b>                              | iv  |
| <b>Contents</b>                              | v   |
| <b>List of Figures</b>                       | vi  |
| <b>List of Abbreviations</b>                 | vii |
| <b>CHAPTER 1 INTRODUCTION</b>                | 1   |
| 1.1. CRYPTOCURRENCIES                        | 1   |
| 1.2. MOTIVATION                              | 2   |
| 1.3. OBJECTIVE                               | 3   |
| 1.4. THESIS STRUCTURE                        | 3   |
| <b>CHAPTER 2 BLOCKCHAIN TECHNOLOGY</b>       | 4   |
| <b>CHAPTER 3 TECHNOLOGY OVERVIEW</b>         | 10  |
| <b>CHAPTER 4 DECENTRALIZED EXCHANGES</b>     | 14  |
| <b>CHAPTER 5 METHODOLOGY</b>                 | 17  |
| 5.1.SWAPPING OF TOKENS                       | 17  |
| 5.2.PROVIDING LIQUIDITY IN TOKEN PAIRS       | 19  |
| 5.3.PROPOSED SOLUTION                        | 21  |
| <b>CHAPTER 6 CONCLUSION AND FUTURE SCOPE</b> | 25  |
| <b>BIBLIOGRAPHY</b>                          | 26  |
| <b>APPENDICE</b>                             | 28  |
| PLAGIARISM REPORT                            | 28  |

## LIST OF FIGURES

| <b>Figure No.</b> | <b>Figure Title</b>   | <b>Page No.</b> |
|-------------------|---|-----------------|
| 2.1               | Structure of a Blockchain                                       | 6               |
| 3.1               | Dex Network and State updating in blockchain                    | 16              |
| 5.1               | Uniswap Constant AMM Curve                                      | 18              |
| 5.2               | Code for Swap function in Uniswap                               | 19              |
| 5.3               | Structure of Liquidity Pool before and after adding token pairs | 20              |
| 5.4               | Code for Adding liquidity function in Uniswap                   | 21              |
| 5.5               | Code for Optimal One sided Liquidity Mining in Uniswap          | 24              |

## LIST OF ABBREVIATIONS

| <b>Abbreviation</b> | <b>Definition</b>       |
|---------------------|-------------------------|
| DeX                 | Decentralised Exchanges |
| CeX                 | Centralised Exchanges   |
| DeFi                | Decentralized Finance   |
| LP                  | Liquidity Provider      |
| UNI                 | Uniswap                 |
| CeFi                | Centralized Finance     |
| AMM                 | Automated market makers |



# CHAPTER 1

## INTRODUCTION

### 1.1 CRYPTOCURRENCIES

The Concept of cryptocurrency started back in 2009 by Satoshi Nakamoto. The first popular cryptocurrency was developed by Satoshi was named as Bitcoin. At that time it was considered useless but nobody at that time had imagined that one day the rate of bitcoin[7] can go upto \$52000. The word cryptocurrency can be divided into two words crypto and currency. It means a currency which is digitally developed and is secured via cryptographic algorithms which makes it impossible to duplicate or double-spend. These Cryptocurrencies are minted in a digital distributed ledger also called as Blockchain. This distributed ledger stores all the transaction that are related to the cryptocurrency and this ledger is distributed across all the nodes that are the part of the blockchain so every node has some rights over all the transactions which makes it decentralized. If a node tries to manipulate a transaction then other nodes will easily identify that and not only the transaction will get rejected but the other nodes will eliminate the malicious node form the network. Satoshi Nakamoto created Bitcoin just for transfer of value or assets just like gold but currently the DeFi industry has a lot of cryptocurrencies which are used for several purposes like medium of payment for storing data, execution of code and also as a 1:1 conversion with the Fiat currency backed by financial institutions. Now a days people have started using cryptocurrencies as a mode of payment due to various reasons like transactional security, anonymity, less transactional fees as it doesn't involve any middleman and works in a trust less, fast and reliable cross border environment. Cryptocurrencies can also be used for investments, buying stocks, purchasing ownership of an entity, paying different utility bills etc. The cryptocurrencies other than bitcoin are called AltCoins. The popular Altcoins are Ethereum which created a revolution as it provides a Ethereum Virtual Machine platform[2] which can be used for running smart contracts i.e.; a set of instructions that executes itself automatically in a blockchain when some conditions are met. The crypto token for Ethereum[8] blockchain is Ether, it is used for paying the transactional fess on the Ethereum blockchain, DAI which is backed by financial institutions and pegged with US dollars by 1:1 ratio, Matic which is the first Indian blockchain network that works on layer 2 Scaling of Ethereum blockchain by using the concept of sidechains where transaction are verified and first and then the hash of all the transactions are moved to the main Ethereum blockchain which makes the user to pay less transaction fees compared with Ethereum

blockchain, UNI which acts as a governance token in Dex which can be used for different kinds of decisions related to Uniswap. There are several protocols that runs on Ethereum blockchain some of them are Uniswap, Aave, Compound etc. The smart contracts are the basic building blocks of these protocols. In this paper we will consider Uniswap protocol as the base to provide solution for optimal one sided liquidity addition. Uniswap provides various features like swapping of different token and liquidity mining using constant product AMM, staking, lending, borrowing. To meet the challenges user has to face in traditional order book models, these AMMs does their work more effectively and currently it is one of the areas of decentralized finance receiving more attention. Unlike in traditional money market models[1], where user needs to wait for a long time for their orders need to match these AMMs takes just the execution time. In these AMM models liquidity providers need to add liquidity for a token pair and in exchange of that they gets trading fees. The liquidity then is managed by different smart contracts deployed on blockchain. The smart contract decides the exchange price between the token pairs using a mathematical curve.

## **1.2 MOTIVATION**

The motivation behind developing a solution for optimal one-sided liquidity mining in a DeX revolves around enhancing liquidity provision, incentivizing participation, addressing impermanent loss concerns, optimizing capital utilization, fostering flexibility and inclusivity, and promoting ecosystem growth. By addressing these motivations, the solution aims to create a more robust and efficient DeX ecosystem that benefits all participants. One-sided liquidity mining aims to attract liquidity providers who prefer to stake a single asset instead of participating in trading pairs. By incentivizing one-sided liquidity provision, the solution can potentially attract a broader range of participants, thereby increasing overall liquidity in the DeX. Improved liquidity depth reduces slippage and enhances trading experiences for users. Impermanent loss is a common concern for liquidity providers in DeXs. By focusing on one-sided liquidity provision, the solution can help mitigate impermanent loss risks associated with providing liquidity in trading pairs. This addresses the concerns of potential liquidity providers and provides a more attractive option for participation.

### **1.3 OBJECTIVE**

The objective of a solution for optimal one-sided liquidity mining in a decentralized exchange (DeX) is to provide a mechanism that encourages liquidity providers to participate by staking a single asset in a way that maximizes their rewards and maintains the efficiency of the liquidity pool. One-sided liquidity mining aims to encourage liquidity providers to stake a single asset instead of providing liquidity in a trading pair. The solution should provide sufficient incentives to attract liquidity providers to participate in this model by offering competitive rewards and benefits. The solution should optimize the rewards for liquidity providers by taking into account factors such as the amount of liquidity provided, the duration of staking[10], and the trading volume generated. It should provide a mechanism to allocate rewards in a way that reflects the contribution of each liquidity provider accurately. Optimal one-sided liquidity mining should aim to efficiently allocate the available capital within the liquidity pool to maximize the liquidity depth and minimize slippage for traders. The solution should ensure that the liquidity pool remains balanced and robust, even with one-sided liquidity provision. The solution should consider the risks associated with one-sided liquidity provision, such as impermanent loss, and provide mechanisms to mitigate or compensate for these risks. It should include risk management strategies that help liquidity providers make informed decisions and manage their exposure effectively.

### **1.4 THESIS STRUCTURE**

The rest of the thesis is divided into the following Chapters.

Chapter 2 consists of the way blockchain technology works, the way blocks are formed and the manner in which the blocks are connected.

Chapter 3 discusses the different technologies that are involved in decentralized finance, cryptocurrencies

Chapter 4 shows the way decentralized exchanges work, the way participants in the decentralized exchanges do their transaction and update of block data.

Chapter 5 includes the actual solution of optimal one-sided liquidity mining in decentralized exchanges. It is divided into three parts namely swapping of tokens, addition of liquidity in a pool and the actual solution

Chapter 6 discusses the conclusion and the future scope of the current work.

## CHAPTER 2

### BLOCKCHAIN TECNOLOGY

Blockchain Technology is a distributed, decentralized and transparent ledger system that allows multiple nodes on a blockchain network to maintain and share large group of transactions in a secured manner. It has gained its popularity from the invention of cryptocurrencies like Bitcoin and Ethereum. It has its potential use cases in various sectors other than finance.

Under the hood, the blockchain consists of multiple blocks linked together using cryptographic algorithms where each block consists majorly of two things i.e., block-header and block-body. The block-body contains all the transaction that are executed and verified by the miner nodes and the block-header has different parameters to identify the characteristic of the block like previous block hash, timestamp, block number, difficulty, gas consumption, gas price, miner rewards and address, nonce and version. The data are stored in the form of chronological ordering of blocks. Blockchain is maintained by all the participating nodes in the network. Each node will make a backup

Key Features:

- **Decentralization:** Blockchain operates in a decentralized manner, meaning that there is no central authority controlling the entire network. Instead, multiple participants (often referred to as nodes) maintain copies of the blockchain and collectively validate transactions.
- **Distributed Consensus:** Consensus mechanisms are used to agree on the state of the blockchain and validate transactions. Various consensus algorithms, such as Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT), ensure that a majority of participants agree on the validity of transactions before they are added to the blockchain.

- **Security and Immutability:** The use of cryptographic hashes and decentralized consensus makes blockchains highly secure and resistant to tampering. Once a block is added to the blockchain, it becomes extremely difficult to alter or delete the data within it.
- **Transparency:** Blockchain provides transparency as all transactions are recorded on the ledger, which can be accessed and verified by anyone on the network. This transparency fosters trust among participants and eliminates the need for intermediaries in many scenarios.
- **Smart Contracts:** Smart contracts are self-executing contracts with predefined rules encoded on the blockchain. They automatically enforce the terms and conditions of an agreement between parties. Smart contracts enable automation, efficiency, and the creation of decentralized applications (DApps) on top of blockchain platforms.
- **Use Cases:** While blockchain's initial application was in the realm of cryptocurrencies, it has since found use in various industries. Examples include supply chain management, healthcare data exchange, financial services, voting systems, intellectual property protection, and more. Blockchain's ability to provide transparency, security, and decentralization has made it attractive for solving trust-related issues in numerous domains.

It's important to note that there are different types of blockchains, such as public blockchains (open to anyone), private blockchains (restricted access), and consortium blockchains (shared among a group of organizations). Each type has its own advantages and use cases. Overall, blockchain technology has the potential to revolutionize industries by providing secure, transparent, and decentralized solutions to various problems. Its widespread adoption and further development continue to shape the future of many sectors.

Blockchain technology has found applications across various industries due to its unique features of decentralization, security, transparency, and immutability. Here are some notable applications of blockchain. The most well-known application of blockchain is cryptocurrencies like Bitcoin and Ethereum. Blockchain enables secure and decentralized digital currencies, eliminating the need for intermediaries like banks for financial transactions. Blockchain can enhance supply chain transparency by tracking and recording the movement of goods from

their origin to the final destination. This ensures the authenticity, provenance, and quality of products, prevents counterfeiting, and improves traceability. Blockchain has the potential to revolutionize traditional financial systems. It can facilitate faster, secure, and cost-effective cross-border transactions, remittances, and peer-to-peer transfers. It also enables the creation of decentralized financial applications (DeFi) that provide services like lending, borrowing, and decentralized exchanges. Blockchain can improve the security and accessibility of electronic health records (EHRs) by providing a tamper-proof and decentralized storage system. It enables patients to have control over their health data, simplifies data sharing between healthcare providers, and enhances interoperability. Blockchain can facilitate supply chain financing by providing transparency and traceability of transactions. It allows businesses to track and verify transactions, improving trust between parties and enabling access to financing based on real-time supply chain data.

These are just a few examples of the wide range of applications for blockchain technology. As the technology continues to evolve and mature, new innovative use cases are likely to emerge in various sectors.

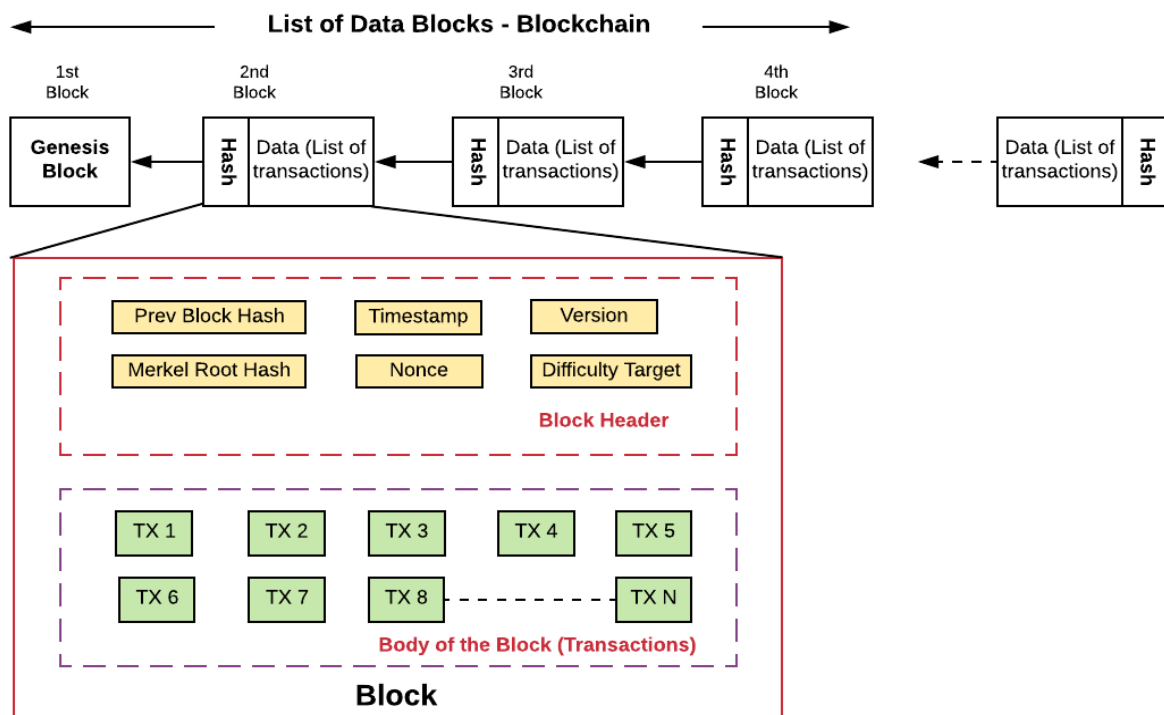


Figure 2.1 :: Structure of a Blockchain

In this figure 2.1, the structure of the blockchain is shown where the genesis block is the initial or first block in a blockchain. It is the foundation upon which the entire blockchain network is

built. The genesis block is typically hardcoded into the blockchain's protocol and serves as a reference point for subsequent blocks in the chain.

Here are a few key aspects of a genesis block:

1. **Creation:** The genesis block is manually created by the creator or developer(s) of the blockchain network. It is the first block added to the chain and does not reference any previous blocks since there are none.
2. **Unique Characteristics:** The genesis block often has unique characteristics that distinguish it from regular blocks. For example, it may have a different structure or contain specific data that identifies it as the genesis block.
3. **Block Hash:** Similar to other blocks in the blockchain, the genesis block has a cryptographic hash associated with it. This hash is usually generated using a hash function, such as SHA-256, and serves to validate the integrity of the block.
4. **Initial Parameters:** The genesis block sets the initial parameters and rules for the blockchain. This includes information such as the network's version, timestamp, difficulty level, and other necessary data to establish the blockchain's initial state.
5. **Distribution of Cryptocurrency:** In blockchain networks that involve cryptocurrencies, the genesis block often includes the initial distribution of coins or tokens. It specifies the initial allocation of cryptocurrency units to specific addresses or entities, including the creator(s) of the blockchain.
6. **Starting Point:** The genesis block acts as the starting point for the blockchain's history. All subsequent blocks are linked back to the genesis block through a chain of cryptographic hashes, forming the complete blockchain.

The genesis block is critical for establishing the foundation of a blockchain. Its integrity and validity are crucial for the trustworthiness and security of the entire network. Once the genesis block is created, subsequent blocks can be added to the chain through the consensus mechanism defined by the blockchain protocol.

A block header is a crucial component of each block in a blockchain. It contains essential information that helps identify, validate, and link blocks together in the blockchain. The block header typically consists of the following components.

- **Version:** The version number indicates the protocol version being used for the block. It allows for upgrades and compatibility with different rules and features.

- **Previous Block Hash:** This field contains the hash value of the header of the previous block in the blockchain. It creates a linkage between blocks, ensuring a sequential and chronological order in the blockchain.
- **Merkle Root:** The Merkle root is a hash of all the transactions included in the block. It is computed by organizing the transaction hashes into a binary tree structure (Merkle tree) and hashing them until a single root hash is obtained. The Merkle root provides a concise representation of all the transactions in the block.
- **Timestamp:** The timestamp records the time when the block was created or mined. It serves as a reference point and helps maintain the chronological order of blocks in the blockchain.
- **Nonce:** The nonce (short for "number used once") is a 32-bit field that miners modify during the mining process. Miners repeatedly change the nonce value until they find a hash that meets certain criteria, such as satisfying the difficulty level set by the blockchain's consensus algorithm. The nonce helps ensure that the block hash meets specific conditions and contributes to the security and immutability of the blockchain.
- **Difficulty Target:** The difficulty target represents the level of difficulty required for miners to solve the cryptographic puzzle and add a new block to the blockchain. It adjusts periodically to maintain a consistent block creation rate, ensuring the network's stability and security.
- **Additional Information:** Depending on the specific blockchain protocol, the block header may contain additional fields. These can include extra nonce values, bits representing the network consensus rules, or any other relevant data required by the blockchain's protocol.

The block header, along with the block's transactions, is hashed together to generate the block's unique identifier, called the block hash. This hash value is crucial for validating the block's integrity and linking it to the previous block, forming a chain of blocks in the blockchain.

By including important information and cryptographic hashes, the block header ensures the integrity, security, and immutability of the blockchain while providing a mechanism for efficient verification and consensus among network participants.

The body of a block in a blockchain typically contains the actual data or payload associated with the block. While the block header provides essential information for identifying and linking blocks, the block body carries the transactions, records, or other relevant data specific



to the blockchain's purpose. The exact structure and content of the block body can vary depending on the blockchain protocol and its intended use cases. Here are a few examples:

- **Transaction Data:** In most blockchain networks, the block body contains a list of transactions. Each transaction represents a transfer of assets, data, or information between participants on the blockchain. Transaction data usually includes sender and recipient addresses, the amount or value transferred, transaction fees, and any additional data specific to the transaction type or smart contract execution.
- **Smart Contracts and Code:** For blockchain platforms that support smart contracts, the block body may include the actual code or bytecode associated with the executed smart contracts. This allows for the execution and validation of smart contracts on the blockchain network.
- **Additional Data:** Depending on the specific blockchain use case, additional data relevant to the network may be included in the block body. For example, in supply chain management blockchains, the block body may contain information about the origin, movement, and quality of goods. In healthcare blockchains, it could include patient data or medical records. This additional data provides context and allows for the specific functionality and purpose of the blockchain.
- **Merkle Tree Data:** While the Merkle root is included in the block header, the actual data structure for the Merkle tree is part of the block body. The Merkle tree organizes the transaction data into a binary tree structure, facilitating efficient verification of transactions and proof of inclusion in the block.

It's important to note that the size and content of the block body can significantly impact the block's size, propagation time, and storage requirements within the blockchain network. Blockchain protocols often implement mechanisms to manage block size limits, optimize data storage, and ensure efficient network performance.

By combining the block header, which includes the block hash and metadata, with the block body containing the actual data or transactions, the blockchain maintains a secure, transparent, and immutable ledger of transactions or records.

## CHAPTER 3

### TECHNOLOGY OVERVIEW

Decentralized Finance (DeFi) is a rapidly growing sector that leverages blockchain technology to provide financial services and applications in a decentralized manner. Here's an overview of the key technologies that power DeFi:

#### 1. Defi vs Cefi

DeFi or Decentralized Finance[4] market gained its popularity because of one core concept of decentralization. It allows any person or entity that may be located at different corners of the world to transfer assets or cryptos without revealing its own identity or without any kind of middleman's like a bank or an financial institutions. All the transactions are made using instructions that runs itself automatically when certain conditions are met. These set of instructions packed together in a contract called smart contract. In these types of institutions no one has the controlling authority the code will act according to some predefined protocols. Defi being an emerging market comes with certain types of risks like the user needs to have technical knowledge about the working of the protocols. There can be code errors that can affect the state of the transactions.

CeFi or Centralized Finance works with the control of central controlling authority like banks and financial institutions. It is easy to use CeFi but the transaction cost is high compared to Defi and less secure than Defi. In CeFi, the user doesn't have full control over its funds and accounts. In worst case the government can take the control over the accounts of a person and can cease the entire funds unlike in DeFi.

#### 2. Dex and Cex

CeX or Centralised Exchanges are the entities that are owned by some central authority. The central authority holds the power and have control over all the funds, transfers and wallets of the user. CeX provides custodial type of wallet in which all the funds and private key are held by that central authority, once server gets maliciously attacked all the funds and wallet will be

compromised. All the existing CeX like Zebpay, WazirX follows a model called order book. The makers put their ask price and amount of tokens in the order book, the algorithm searches for the takers bid and if matched the transfer of funds takes place. In CeX, a user can buy cryptos by using fiat currency, those fiat currencies need to be transferred into the exchange's wallet and need to be claimed in the application's wallet by providing certain proofs.

While on the other hand DeX or decentralized exchanges lets user to swap between tokens using liquidity pools. There are no central authority who manages these Dex, these Dex are controlled by instructions that runs itself automatically when certain conditions are met. These set of instructions packed together in a contract called smart contract. The Dex uses mathematical algorithms known as AMM for exchange of tokens. The user need not have to wait for a long time unlike in CeX , here already a pool of token pairs will be available and the exchange rate is decided by the AMM. In Dex we use non custodial wallets, means the private keys and the tokens will be there with us and nobody can have the control over that except the user.

### **3. Uniswap AMM**

The principal component the drives the Uniswap[12] Dex is AMM or Automated Market Makers. AMM's are the mathematical functions that are constructed and implemented using smart contracts which are the basic building blocks of the DeX. Uniswap uses a constant product market maker algorithm,  $x*y=k$  which not only decide the exchange rate of every Ethereum based token but also balances the liquidity pool according to the supply and demand.

### **4. Token Swapping**

Token swapping is process of exchanging one token with another token in a trustless environment and this will happen in one transaction only which will make it atomic in nature. In swapping both maker and taker will receive the tokens or the transaction will be cancelled. The token swapping is controlled by a timelock smart contract in which the exchange rate is varied time to time with respect to the current liquidity. Sometimes the token swapping multiple timelock contract interact with each other if the pair liquidity is not actively present on that pool.

## **5. Liquidity Mining**

Liquidity Mining[11] is a process of providing tokens in pairs to a liquidity pool[3] in decentralized exchanges. It is practically not feasible for a Dex organisation to provide liquidity pools with different token pairs. So it needs some other entity to provide liquidity to its pool and in return that DeX reward that entity with proportional amount of liquidity provider fee that it takes during token swapping.

## **6. Slippage tolerance**

In DeX while swapping of tokens used to take place there will be a small timing gap between the initiation of the transaction and acceptance of the transaction. In the blockchain the transactions are processed one after the another so that may affect the state of the liquidity pool from which the swap is taking place. It can be due to high market volatility or due to less liquidity than the desired one for swap. The difference between the amount of token that uniswap promises to provide while initiating the transaction and the amount of actual number of tokens that we gets after the confirmation of the transaction is called slippage tolerance. If uniswap promises to give 1 MATIC before the initiation of the transaction but after the transaction gets confirmed we gets only 0.75 MATIC then there will be a slippage tolerance of 25%. In most of the DeX the slippage tolerance level is 5% by default but it can be changed while initiating the transaction.

## **7. Smart Contracts**

Smart Contracts[6] are set of instructions that executes automatically when some conditions are met. It helps to provide some set of irreversible agreement between two or more entities. The transactions that are executed inside a smart contract are traceable and is verified by more than 51% of the nodes in the blockchain network and only after that the final state change happens. It is near to impossible to hack smart contracts at blockchain level not on application level. Smart Contracts are the building blocks for the establishment and popularity of the Decentralized Finance market.

## **8. Dapps**

Dapps or Decentralized applications[5] are almost similar to the normal client-server type of applications but unlike traditional application they work in a peer to peer network where all the execution is done using a smart contract. Execution of a function using a dapp is called transaction. All these transactions are irreversible and needs to be verified by maximum nodes of a blockchain in which it is executing. After the verification only we can say a transaction is successfully executed. In maximum of the blockchain we need to provide a small fee so that the transaction can be executed and verified quickly, the fee is called gas fees. The waiting time for the confirmation of the transaction directly proportional to the gas fees.

## CHAPTER 4

### Decentralized Exchanges

Decentralized exchanges (DEXs) are cryptocurrency trading platforms that operate without a central authority or intermediary. Unlike traditional exchanges where transactions are facilitated by a central party, DEXs enable direct peer-to-peer trading through the use of blockchain technology and smart contracts.

In a DEX, users retain control over their funds as they connect their digital wallets directly to the exchange platform. This eliminates the need to deposit funds into a centralized exchange's wallet, reducing the risk of hacks or funds being controlled by a third party.

One popular type of DEX mechanism is the Automated Market Maker (AMM). AMMs are designed to provide liquidity to the decentralized exchange by utilizing liquidity pools instead of traditional order books. Liquidity providers deposit funds into these pools, allowing users to trade against the pool's reserves. These pools are governed by smart contracts that automatically execute trades based on predetermined mathematical formulas.

AMMs use a pricing algorithm, such as the constant product formula, to determine token prices and ensure supply and demand equilibrium within the liquidity pool. When a trade occurs, the smart contract adjusts the pool's reserves proportionally, maintaining the token price ratio.

The advantages of DEXs and AMMs include increased user control over funds, privacy, security, and the ability to trade a wide range of cryptocurrencies. DEXs also foster innovation by enabling the listing and trading of new tokens without the need for approval from a centralized authority.

However, DEXs may face challenges in terms of liquidity compared to centralized exchanges due to their smaller user bases. Nevertheless, the DeFi ecosystem is continuously evolving, and solutions like liquidity pools and decentralized liquidity aggregators are being developed to enhance liquidity on DEXs.

In summary, DEXs and AMMs are at the forefront of decentralized finance, providing a secure and transparent way for users to trade cryptocurrencies directly without relying on intermediaries. Their innovative design and features contribute to the growth and democratization of the cryptocurrency ecosystem.

Here are some key features and characteristics of decentralized exchanges:

1. **Decentralization:** DEXs aim to eliminate the need for a central authority or intermediary. Instead, they leverage smart contracts and blockchain technology to enable direct peer-to-peer trading.
2. **Control of Funds:** In a DEX, users have control over their funds as they typically connect their digital wallets directly to the exchange platform. This eliminates the need to deposit funds into a centralized exchange's wallet, reducing the risk of hacks or funds being controlled by a third party.
3. **Privacy and Anonymity:** DEXs often prioritize user privacy by allowing users to trade without the need for extensive identity verification or KYC (Know Your Customer) processes. This aspect appeals to individuals who value anonymity in their cryptocurrency transactions.
4. **Security:** Since DEXs operate on blockchain networks, they inherit the security features of the underlying technology. Transactions are secured through cryptographic techniques, and the use of smart contracts reduces the risk of human error or manipulation.
5. **Liquidity:** One challenge faced by some DEXs is liquidity. Compared to centralized exchanges, which often have higher trading volumes and liquidity due to their larger user bases, DEXs may face liquidity constraints. However, various initiatives and protocols are being developed to address this issue, such as liquidity pools and decentralized liquidity aggregators.
6. **Trading Options:** DEXs typically support a wide range of cryptocurrencies and tokens based on the blockchain network they operate on. Users can trade directly between different token pairs without the need for intermediaries or centralized order books.
7. **Examples of DEXs:** Some popular decentralized exchanges include Uniswap, SushiSwap, PancakeSwap, and 1inch. Each DEX has its unique features and operates on different blockchain networks, such as Ethereum, Binance Smart Chain, and others.

It's important to note that while DEXs offer various advantages, they may also have limitations, such as scalability challenges, potential front-running attacks, and the inability to interact directly with traditional financial systems. The development of decentralized finance (DeFi)

continues to address these challenges and expand the capabilities of DEXs. The figure 4.1, describes how the participants of the blockchain does their transaction and the updation of the state in the blockchain is shown.

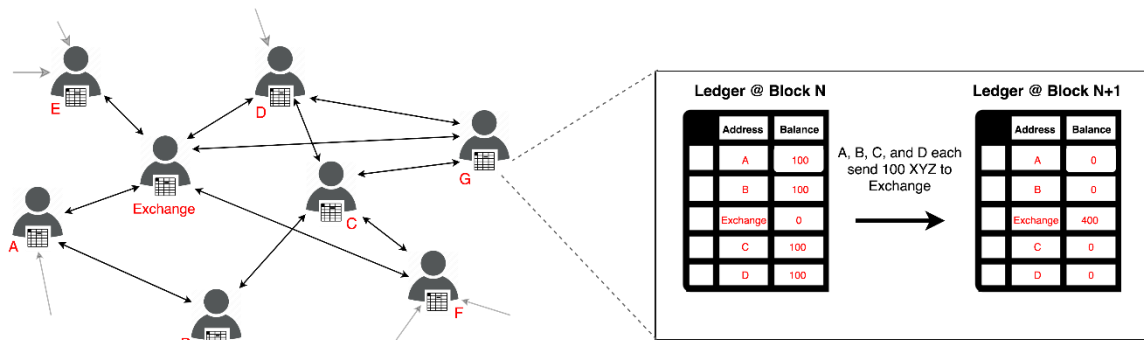


Figure 4.1 :: Dex Network and State updating in blockchain



## CHAPTER 5

### METHODOLOGY

In this paper we are going to introduce the concept of providing one sided liquidity to a liquidity pool in an optimal way so that we can minimize the unutilized token of a liquidity provider. The concept is divided into three parts i) Swapping of tokens using constant product market makers ii) Providing liquidity in token pairs iii) Providing optimal one sided liquidity to a liquidity pool. For every concept we will provide proper mathematical explanations, diagrams and smart contract functions. All the functions will be written based on router contract, pair contract of Uniswap V2 protocol.

#### 1. Swapping of tokens

With the introduction of Uniswap Protocol, the process of swapping of tokens has completely changed. Now a user doesn't need to wait for its order to get approved in the list of order book. Uniswap Protocol is built keeping in mind the benefits of community. Unlike other crypto exchange platforms, uniswap doesn't impose any kind of platform fees. It provides a simple, reliable and secured way of transferring crypto currencies without any middleman and without connecting the makers and takers to decide the exchange rate so that the exchange rate in the market stays stable. Uniswap Protocol has introduced a constant product Automated Market Maker which decides the exchange rate of a token.

The constant product market maker used in Uniswap Protocol is

$$X * Y = K(\text{constant}) \dots \dots \dots (i)$$

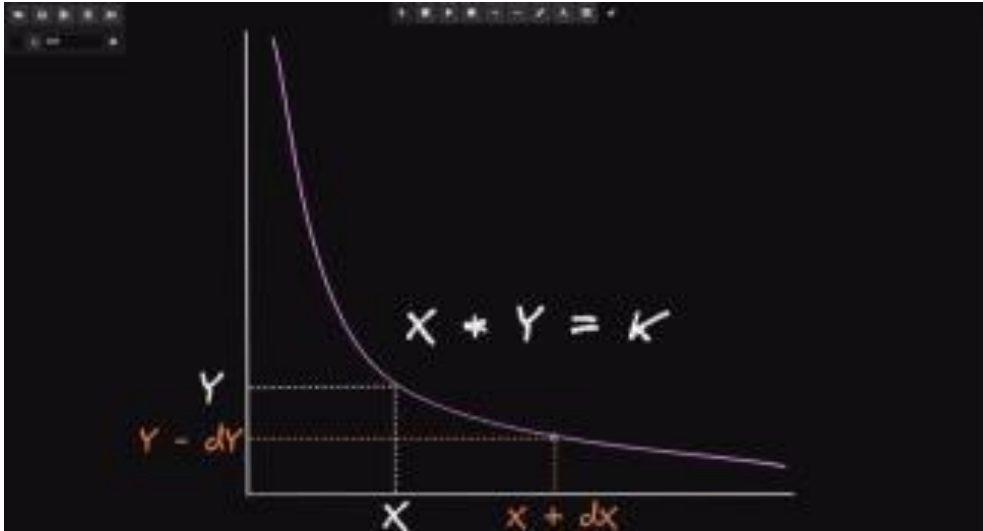


Figure 5.1 :: Uniswap Constant AMM Curve

In the figure 5.1, the plot shows the constant product AMM used in Uniswap. The constant  $K$  is decided by the liquidity pool provider by providing the liquidity of the token pegged by another token. For example, I have created 100 of token Alpha and I want to provide the liquidity into uniswap so that other people can buy the token  $X$  in exchange of token Beta. So I have to provide the liquidity in token pair of Alpha and Beta and have to create the liquidity pool, suppose I put 100 of token alpha and 100 of token beta, so the  $K$  value will be 10000. As different people will collect token alpha by swapping token beta the exchange rate of alpha with beta will change from time to time but the value of  $K$  will remain same as 10000 until and unless some other liquidity providers adds or removes their liquidity.

In the figure 5.1, we have  $X$  amount of one token and  $Y$  amount of another token. When the user will swap between the tokens the current pool will have  $X+dX$  tokens and  $Y-dY$  tokens. The user has swapped  $dX$  amount of one token with  $dY$  amount of another token. According to the constant product market maker.

$$(X+dX)*(Y-dY)=K.....(ii)$$

$$dY=Y-K/(X+dX) .....(iii)$$

$$dY=Y-XY/(X+dX) \text{ (Because } X*Y=K\text{)}.....(iv)$$

$$dY=(XY+dX*Y-XY) /(X+dX).....(v)$$

$$dY=dX*Y/(X+dX) \dots\dots\dots(vi)$$

The amount of token we will get by swapping another token will depend on the above equation. The figure 5.2, describes the code to implement the swap function between two Ethereum based tokens by giving expected amount of input and output tokens, in this function the eq(vi) is been implemented.

```

// this low-level function should be called from a contract which performs important safety checks
function swap(uint amount0out, uint amount1out, address to, bytes calldata data) external lock {
    require(amount0out > 0 || amount1out > 0, 'Pancake: INSUFFICIENT_OUTPUT_AMOUNT');
    (uint112 _reserve0, uint112 _reserve1,) = getReserves(); // gas savings
    require(amount0out < _reserve0 && amount1out < _reserve1, 'Pancake: INSUFFICIENT_LIQUIDITY');

    uint balance0;
    uint balance1;
    { // scope for _token{0,1}, avoids stack too deep errors
        address _token0 = token0;
        address _token1 = token1;
        require(to != _token0 && to != _token1, 'Pancake: INVALID_TO');
        if (amount0out > 0) _safeTransfer(_token0, to, amount0out); // optimistically transfer tokens
        if (amount1out > 0) _safeTransfer(_token1, to, amount1out); // optimistically transfer tokens
        if (data.length > 0) IPancakeCallee(to).pancakeCall(msg.sender, amount0out, amount1out, data);
        balance0 = IERC20(_token0).balanceOf(address(this));
        balance1 = IERC20(_token1).balanceOf(address(this));
    }
    uint amount0In = balance0 > _reserve0 - amount0out ? balance0 - (_reserve0 - amount0out) : 0;
    uint amount1In = balance1 > _reserve1 - amount1out ? balance1 - (_reserve1 - amount1out) : 0;
    require(amount0In > 0 || amount1In > 0, 'Pancake: INSUFFICIENT_INPUT_AMOUNT');
    { // scope for reserve{0,1}Adjusted, avoids stack too deep errors
        uint balance0Adjusted = (balance0.mul(10000).sub(amount0In.mul(25)));
        uint balance1Adjusted = (balance1.mul(10000).sub(amount1In.mul(25)));
        require(balance0Adjusted.mul(balance1Adjusted) >= uint(_reserve0).mul(_reserve1).mul(10000**2), 'Pancake: K');
    }

    _update(balance0, balance1, _reserve0, _reserve1);
    emit Swap(msg.sender, amount0In, amount1In, amount0out, amount1out, to);
}

```

Figure 5.2:: Code for Swap function in Uniswap

## 2. Providing Liquidity in Token Pairs

Liquidity acts as a fuel in crypto DeX industry. Uniswap needs to have sufficient liquidity so that it can provide its basic functionality of swaps. Currently providing liquidity into a liquidity pool is one of the most profitable investment in DeX. Every user can provide liquidity to a pool with any amount of token pairs and the rewards will get distributed accordingly. During swapping of tokens a small amount of fee is taken, which we call as a liquidity provider fee that is distributed over all the liquidity providers according to the proportion of their liquidity investment. A person’s liquidity can be taken out at any point of time using LP tokens, which gives the person a sense of freedom financially. The way and the amount of token pairs that will be added in the liquidity pool will be decided by constant

product market of Uniswap Protocol. A person cannot add any amount of token pairs in the liquidity pool. Uniswap balances the pool by allowing an user to add liquidity of the token pairs proportional to that of the pool's token pairs.

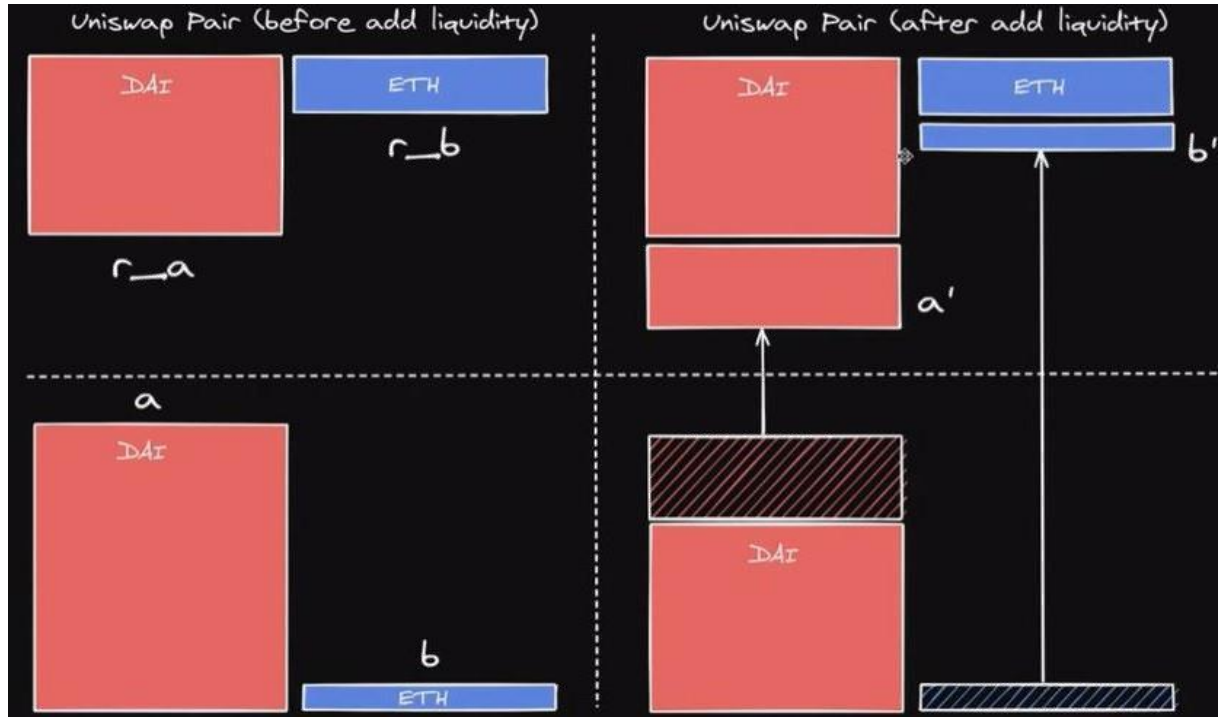


Figure 5.3:: Structure of Liquidity Pool before and after adding token pairs

In the figure 5.3, the plot is divide into two parts i) Before proving liquidity and ii) After providing liquidity. The above portion signifies amount of token pair that uniswap have currently and the below one signifies amount of token pair that the liquidity provider have. Currently the liquidity pool is balanced with  $r_a$  amount of crypto DAI and  $r_b$  amount of crypto ETH. The user has  $a$  amount of DAI and  $b$  amount of ETH. Now the user can add the liquidity in the ratio  $r_a:r_b$  so that the pool remain balanced. So one of the cryptos of the user will be fully utilized and the some part of the another token will be reverted back to the user. According to the diagram out of  $a$  tokens of DAI only  $a'$  is accepted by the pool and complete  $b$  tokens of ETH is accepted which we have indicated by  $b'$ . Now we are going to calculate the amount of tokens that will be utilized while a user will provide liquidity.

$$r_a:r_b = (r_a+a'):(r_b+b') \dots\dots\dots(vii)$$

$$r_a*(r_b+b') = (r_a+a')* r_b \dots\dots\dots(viii)$$

$$r_a*b' = r_b*a' \dots\dots\dots(ix)$$

If  $a'=a$  ,then  $b>b'$  therefore  $b'=(r_b/r_a)*a'$

If  $b'=b$  ,then  $a>a'$  therefore  $a'=(r_a/r_b)*b'$

```
// **** ADD LIQUIDITY ****
function addLiquidity(
  address tokenA,
  address tokenB,
  uint amountADesired,
  uint amountBDesired,
  uint amountAMin,
  uint amountBMin
) internal virtual returns (uint amountA, uint amountB) {
  // create the pair if it doesn't exist yet
  if (IPancakeFactory(factory).getPair(tokenA, tokenB) == address(0)) {
    IPancakeFactory(factory).createPair(tokenA, tokenB);
  }
  (uint reserveA, uint reserveB) = PancakeLibrary.getReserves(factory, tokenA, tokenB);
  if (reserveA == 0 && reserveB == 0) {
    (amountA, amountB) = (amountADesired, amountBDesired);
  } else {
    uint amountBOptimal = PancakeLibrary.quote(amountADesired, reserveA, reserveB);
    if (amountBOptimal <= amountBDesired) {
      require(amountBOptimal >= amountBMin, 'PancakeRouter: INSUFFICIENT_B_AMOUNT');
      (amountA, amountB) = (amountADesired, amountBOptimal);
    } else {
      uint amountAOptimal = PancakeLibrary.quote(amountBDesired, reserveB, reserveA);
      assert(amountAOptimal <= amountADesired);
      require(amountAOptimal >= amountAMin, 'PancakeRouter: INSUFFICIENT_A_AMOUNT');
      (amountA, amountB) = (amountAOptimal, amountBDesired);
    }
  }
}
}
```

Figure 5.4:: Code for Adding liquidity function in Uniswap

The figure 5.4, describes the code to implement the add liquidity function using two Ethereum based tokens by giving desired amount of input tokens, token address and minimum amount of input tokens and in this function the  $eq(ix)$  is been implemented.

### 3) Providing optimal one sided liquidity to a liquidity pool.

The core concept of this paper is to provide a solution for providing liquidity to a pool using a single token so that the amount of token that will be returned while balancing the pool can be reduced as much as possible. This solution will combine the above two solutions so that the objective of this paper can be met. The mathematical explanation will demonstrate how can we add one sided liquidity to a liquidity pool in a optimal way.

Before starting solving mathematical equations we need to define some variables:

A : amount of token A in Uniswap

B : amount of token B in Uniswap

f : trading fee / liquidity provider fee

a : amount of token A I have

b : amount of token B I need

s : amount of token to swap from A => B

So now our objective is to identify b and s so that the liquidity supply will be optimal

AIM : b=? and s=?

We know uniswap uses constant product market maker algorithm i.e.;  $AB=K(\text{constant})$

$$K=AB \dots\dots\dots(x)$$

If I swap s amount of token A then the pool will get only  $(1-f)*s$  tokens because of liquidity provider fee. The amount of token I will get in return is b. As the pool will be balanced by considering in the mean time no other liquidity is added or removed.

$$K = (A+(1-f)s) * (B-b) \dots\dots\dots(xi)$$

$$b = B(1-f)s / (A+(1-f)s) \dots\dots\dots(xii)$$

As we know the pool is balanced the ratio of token pairs before swapping will be same and liquidity will be added on that ratio only so,

$$(A+s) / (B-b) = (a-s) / b \dots\dots\dots(xiii)$$

$$(A+s)b - (a-s)(B-b) = 0 \dots\dots\dots(xiv)$$

$$Ab + sb - aB + ab + sB - sb = 0 \dots\dots\dots(xv)$$

$$Ab - aB + ab + sB = 0 \dots\dots\dots(xvi)$$

Substituting the value of b from equation (i)

$$(A+a)B(1-f)s / (A+(1-f)s)-(a-s)B = 0 \dots\dots\dots(xvii)$$

$$(A+a)(1-f)s / (A+(1-f)s)-(a-s) = 0 \dots\dots\dots(xviii)$$

$$(A+a)(1-f)s - (a-s) (A+(1-f)s) = 0 \dots\dots\dots(xix)$$

$$(A+a)(1-f)s - aA - a(1-f)s + sA + (1-f)s^2 = 0 \dots\dots\dots(xx)$$

$$A(1-f)s - aA + sA + (1-f)s^2 = 0 \dots\dots\dots(xxi)$$

$$A(2-f)s - aA + (1-f)s^2 = 0 \dots\dots\dots(xxii)$$

$$(1-f)s^2 + A(2-f) - aA = 0 \dots\dots\dots(xxiii)$$

This is a quadratic equation and we need to solve for s to get the optimal value of b

For  $ax^2 + bx + c = 0$ , the roots will be  $x = (-b \pm \sqrt{b^2 - 4ac}) / 2a$

For our equation we get the value of s as  $(-(2-f) \cdot A + \sqrt{((2-f) \cdot A)^2 + 4(1-f) \cdot Aa}) / 2(1-f)$

In Uniswap the value of f is 0.3% so the resultant s will be

$$s = (-1997A + \sqrt{3988009A^2 - 3988000 Aa}) / 1994 \dots\dots\dots(xxiv)$$

```

41 function zap(
42     address _tokenA,
43     address _tokenB,
44     uint _amountA
45 ) external {
46     IERC20(_tokenA).transferFrom(msg.sender, address(this), _amountA);
47
48     address pair = IUniswapV2Factory(FACTORY).getPair(_tokenA, _tokenB);
49     (uint reserve0, uint reserve1, ) = IUniswapV2Pair(pair).getReserves();
50
51     uint swapAmount;
52     if (IUniswapV2Pair(pair).token0() == _tokenA) {
53         // swap from token0 to token1
54         swapAmount = getSwapAmount(reserve0, _amountA);
55     } else {
56         // swap from token1 to token0
57         swapAmount = getSwapAmount(reserve1, _amountA);
58     }
59
60     _swap(_tokenA, _tokenB, swapAmount);
61     _addLiquidity(_tokenA, _tokenB);
62 }
63 }

```

```

36 */
37 function getSwapAmount(uint r, uint a) public pure returns (uint) {
38     return (sqrt(r.mul(r.mul(3988009) + a.mul(3988000))).sub(r.mul(1997))) / 1994;
39 }
40 }
41

```

Figure 5.5:: Code for Optimal One sided Liquidity Mining in Uniswap

The figure 5.5, describes the code to implement optimal one sided liquidity mining where it takes the input token addresses so that the user can put their liquidity in that pool along with that we need the token amount of a single token. The above code implements the logic of the equation (xxiv) and calls two other internal functions like swap and add liquidity.



## CHAPTER 6

### CONCLUSION AND FUTURE SCOPE

#### 1. Conclusion

With the advancements in crypto industry, creating different types of DeFi protocols has become a very hot research topic. Everyday one or the other is working on building different protocols or securing and optimising the currently running protocols. Currently few minds are focusing on to reduce the transactions confirmation time and high transactional fees. One of the successful solution that polygon has introduced is to use sidechains for the confirmation of the transactions then to put the combined confirmed hashes to the main Ethereum blockchain. In this paper I have successfully given the mathematical proof and sample code for the optimal supply of providing one sided liquidity in Uniswap Protocol. It will help the liquidity providers to get maximum profit from the liquidity mining as well as the AMM's to get sufficient amount of liquidity for the use of swapping of tokens.

#### 2. Future Scope

The future scope of AMMs in DEXs involves continuous improvements in liquidity, capital efficiency, interoperability, price oracles, governance, and incentivization mechanisms to create a more efficient and user-friendly trading experience. The improvements are speculative and based on current trends and potential advancements in the decentralized exchange space. The actual direction and development of Uniswap will depend on various factors, including market demands, technological advancements, and community consensus. The concept of concentrated liquidity, which is a notable feature that offers more control and flexibility to liquidity providers.



[9] Kanan Arora (2021). Centralized Exchange. Updated Aug 06, 2021, Accessed on December 9, 2021. [Online].

Available:<https://www.analyticssteps.com/blogs/centralized-and-decentralized-cryptocurrency-exchanges>

[10] Minyarinn Chaotrakul (2019) Staking. Updated Nov 11, 2019, Accessed on December 9, 2021. [Online]. Available:<https://medium.com/bitkub/staking-and-cryptocurrency-c762ff8daf5d>

[11] M Glenn Bonez Bona (2021), Liquidity Mining(2021) Updated May 10, 2021. Accessed on December 9, 2021. [Online]. Available:<https://www.matrixswap.io/blog/what-is-liquidity-mining>

[12] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, Dan Robinson(2021), Uniswap Protocol v3(2021) [Online]. Available : <https://uniswap.org/whitepaper-v3.pdf>



PAPER NAME

**Report for plag.pdf**

WORD COUNT

**7276 Words**

CHARACTER COUNT

**39256 Characters**

PAGE COUNT

**28 Pages**

FILE SIZE

**674.4KB**

SUBMISSION DATE

**May 30, 2023 11:11 AM GMT+5:30**

REPORT DATE

**May 30, 2023 11:11 AM GMT+5:30**

● **13% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 6% Internet database
- 5% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

● **Excluded from Similarity Report**

- Bibliographic material