POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea

Finanza Decentralizzata: strategie per la gestione dei Liquidity Pool

Relatore
Prof. Danilo Bazzanella

Candidato
Carlotta Agnese Trovati

In collaborazione con
Links Foundation

Dott. Enrico Ferro
Dott. Roberto Moncada

Anno Accademico 2024-2025

Sommario

Nel contesto della finanza decentralizzata applicata alla tecnologia blockchain, i liquidity pool rappresentano un elemento chiave per il funzionamento degli exchange decentralizzati e la gestione della liquidità.
Questa tesi dunque ha l'obiettivo di analizzare il processo decisionale che porta alla creazione di un liquidity pool, con particolare attenzione alla scelta del setup ottimale e alla definizione degli incentivi per i fornitori di liquidità. Inizialmente, vengono analizzati i concetti fondamentali alla base dei liquidity pool, concentrandosi sui diversi tipi di market maker automatizzati. L'elaborato, successivamente, si concentra sui fattori critici da considerare in fase di apertura di un liquidity pool, quali la scelta degli asset, il rapporto tra i token, la gestione della volatilità e l'impatto delle fee sulle strategie degli investitori. Viene discusso il concetto di impermanent loss, evidenziando le cause e i metodi che possono essere utilizzati per ridurne l'impatto. A tale scopo si analizza il processo decisionale da seguire per la scelta delle caratteristiche ottimali del pool e degli incentivi più adatti al contesto di riferimento.
Ci si concentra sulla costruzione di un modello matematico che, dal punto di vista dello sviluppatore di liquidity pool, consenta di ottimizzare tale configurazione anche a livello numerico. Vengono esaminate diverse combinazioni di capitale e commissioni per valutare il comportamento del modello in vari scenari di mercato. L'analisi dei dati raccolti attraverso questo studio fornisce indicazioni pratiche per la progettazione di un liquidity pool.
L'obiettivo finale è favorire una maggiore comprensione di un settore in rapida evoluzione, fornendo informazioni utili per agenti economici che vogliono intraprendere processi di digitalizzazione per mezzo di tecnologie decentralizzate, ma anche un quadro complessivo per gli investitori attivi in mercati finanziari senza intermediari.

Acronimi

AMM Automated Market Maker.

BAMM Bootstrap Automated market maker.

BFT Byzantine Fault Tolerance.

CEX Exchange Centralizzato.

CMMM Automated market maker a media costante.

CPMM Automated market maker a prodotto costante.

CSMM Automated market maker a somma costante.

DeFi Finanza Decentralizzata.

DEX Exchange Decentralizzato.

DLT Distributed Ledger Technology.

HFMM Automated market maker ibrido.

IL Impermanent Loss.

LP Liquidity Provider.

PoS Proof of Stake.

PoW Proof of Work.

Indice

Elenco delle tabelle 6

Elenco delle figure 7

1 Introduzione 9

2 Blockchain 13
2.1 Nascita della blockchain . . . 13
2.2 Basi della blockchain . . . 14
2.3 Algoritmi di consenso . . . 18
2.3.1 Byzantine Fault Tolerance . . . 21
2.3.2 Proof of Work . . . 23
2.3.3 Proof of Stake . . . 24
2.4 Tipologie di blockchain . . . 26
2.5 Token e Smart Contract . . . 28
2.6 Finanza decentralizzata . . . 31

3 Liquidity Pool 35
3.1 Tipologie di AMM . . . 37
3.1.1 AMM a prodotto costante . . . 37
3.1.2 AMM a somma costante . . . 41
3.1.3 AMM a media costante . . . 43
3.1.4 AMM ibridi . . . 43
3.1.5 AMM a liquiidtà concentrata . . . 45
3.1.6 Bootstrap AMM . . . 47
3.2 Impermanent loss . . . 49

4 Obiettivi e metodologia 55
4.1 Processo decisionale . . . 56
4.1.1 Incentivi . . . 61
4.2 Modello di ottimizzazione . . . 63
4.2.1 Forma funzionale liquidità . . . 66

5 Analisi dei risultati 75
5.1 Estensioni future . . . 81

6 Conclusione 83

A Codice per l'estrazione della liquidità 91

B Codice per l'estrazione del volume 95

C Codice per l'analisi di regressione della liquidità 97

D Matrici di correlazione 101

E Codice per la risoluzione del modello 103

F Grafici dei risultati 109

Elenco delle tabelle

2.1 Esempio struttura dell'header di un blocco Bitcoin 19
3.1 Quantità token nel liquidity pool 40
4.1 Pro e contro per ogni coppia di token 57
5.1 Divisione ottimale con incentivi per early adopter 76
5.2 Divisione ottimale con incentivi di lockup 76

Elenco delle figure

2.1 Rappresentazione degli eventi su linea del tempo . . . 14
2.2 Funzionamento crittografia asimmetrica[1] . . . 16
2.3 Differenza tra centralized e distributed ledger[2] . . . 18
2.4 Merkle Tree . . . 20
2.5 Trilemma della Blockchain . . . 21
2.6 Problema dei generali bizantini[3] . . . 22
2.7 Processo seguito dalla PoW [Abukari et al., 2023] . . . 24
2.8 Processo seguito dalla PoS[4] . . . 25
2.9 Differenza tra contratti tradizionali e smart contract [5] . . . 30
2.10 Utenti DeFi nel tempo . . . 32
2.11 Capitalizzazione in dollari della DeFi — Maggio 2020-Giugno 2021 . . . 33
3.1 Grafico rappresentante la formula dei CPMM[6] . . . 38
3.2 Grafico rappresentante la formula dei CSMM[7] . . . 42
3.3 Grafico rappresentante la formula dei HFMM[8] . . . 44
3.4 Rappresentazione dell'efficienza del capitale [Heimbach et al., 2022] . . . 46
3.5 Differenza tra prodotto costante e liquidità concentrata [9] . . . 47
3.6 Ribilanciamento bootstrap AMM . . . 48
3.7 Grafico dell'IL per diverse pool [10] . . . 51
3.8 Diversa IL per intervalli di investimento [Heimbach et al., 2022] . . . 52
4.1 Diagramma di flusso relativo al setup . . . 60
4.2 Diagramma di flusso relativo agli incentivi . . . 62
4.3 Suddivisione del trading sulla piattaforma Uniswap . . . 67
4.4 Frequenza dei token nelle pool di Uniswap . . . 68
4.5 Regressione log-log — $R^2 = 0.83$ . . . 69
4.6 Effetto degli incentivi per early adopter . . . 72
4.7 Incentivi lockup . . . 73
5.1 Risultati con incentivi per early adopter . . . 78
5.2 Matrice di correlazione con incentivi per early adopter . . . 79
5.3 Risultati con incentivi di lockup . . . 80
5.4 Matrice di correlazione con incentivi di lockup . . . 81
D.1 Matrice di correlazione dati originali . . . 101
D.2 Matrice di correlazione dati logaritmici . . . 101
F.1 Risultati con incentivi per early adopter su periodo ristretto . . . 109
F.2 Risultati con incentivi di lockup su periodo ristretto . . . 110

Capitolo 1

Introduzione

Negli ultimi anni, la tecnologia blockchain ha acquisito un ruolo sempre più centrale nel settore finanziario, tecnologico e accademico, offrendo un nuovo paradigma basato su sicurezza, trasparenza e decentralizzazione. Grazie alla sua struttura distribuita e immutabile, la blockchain ha permesso la creazione di nuovi sistemi economici, in cui le criptovalute e i contratti intelligenti sono i protagonisti. Ciò ha favorito la nascita della **finanza decentralizzata** (DeFi), che eliminando la necessità di intermediari, garantisce agli utenti maggiore autonomia nella gestione dei propri asset digitali. Oltre al settore finanziario, la tecnologia blockchain sta trovando applicazione in numerosi ambiti, aprendo nuove prospettive per l'innovazione in diversi settori.
La crescente adozione della DeFi, tuttavia, ha portato alla luce alcune criticità che necessitano di essere approfondite: uno dei principali problemi riguarda la gestione della liquidità e l'efficacia degli incentivi all'interno dei liquidity pool. Questi strumenti, infatti, eliminano la necessità di intermediari centralizzati utilizzando gli **automated market maker** (AMM), ma introducono sfide legate alla stabilità del mercato, alla sicurezza e alla gestione dei rischi finanziari, come l'**impermanent loss** (IL).
Questa tesi, dunque, mira a fornire una panoramica dei concetti fondamentali della blockchain e delle sue applicazioni nel settore finanziario, con l'obiettivo di contribuire alla comprensione delle dinamiche alla base di questa rivoluzione tecnologica e finanziaria. Si pone particolare attenzione alle caratteristiche dei liquidity pool per comprendere come queste rappresentino scelte cruciali in fase di progettazione. A tal fine si vogliono mettere in luce quali siano i principi e i passaggi fondamentali per progettare un liquidity pool che garantisca sostenibilità economica nel lungo periodo. Ciò concentrandosi su quali siano le strategie e meccanismi di incentivazione possono essere adottati per attrarre e mantenere i **liquidity provider** (LP) all'interno di un liquidity pool. A questo scopo viene introdotto un modello di ottimizzazione in grado di seguire gli sviluppatori in questo processo, che dia indicazioni puntuali sulla strategia da adottare.
Questo studio si inserisce in un contesto di ricerca recente che, nonostante ciò, è stato oggetto di diverse analisi sotto molteplici aspetti. Esistono, dunque, studi che hanno analizzato il funzionamento degli AMM, il loro impatto sul mercato e le strategie per migliorarne l'efficienza e la sicurezza. In particolare, la letteratura esistente si è concentrata su diverse prospettive, tra cui le basi matematiche dei modelli AMM, le innovazioni

tecniche introdotte per ottimizzare il loro funzionamento, le sfide operative che i fornitori di liquidità devono affrontare e le strategie per mitigare le perdite impermanenti.
Sono numerosi, infatti, i paper che si soffermano sulla struttura matematica degli AMM, permettendo di modellarne il comportamento e studiarne le dinamiche di prezzo e liquidità [Angeris et al., 2022, Bichuch and Feinstein, 2022, He et al., 2024b, Port and Tiruviluamala, 2022]. Molte di queste analisi, inoltre, si sono concentrate sugli AMM a liquidità concentrata, che hanno rappresentato un'importante innovazione in questo ambito [Fritsch, 2021, Hasbrouck et al., 2023, Heimbach et al., 2022]. Risulta analoga l'analisi svolta in Wang [2020], in cui però vengono introdotte forme di AMM innovativi che si basano su funzioni ellittiche. In Othman and Sandholm [2011] vengono estesi gli AMM tradizionali introducendo un modello con perdita limitata e *spread bid/ask* adattivo, risolvendo problemi chiave di sostenibilità economica e rendendo i market maker algoritmici più simili a quelli tradizionali. Un'ulteriore prospettiva teorica sugli AMM è fornita da Jensen et al. [2021], che propone un approccio universale alla fornitura di liquidità e dimostra l'equivalenza teorica tra *constant function market maker* e *token swap market maker* con riserve uniformi, i quali regolano i prezzi all'interno del pool con funzioni differenti. In Krishnamachari et al. [2021] si introduce il concetto di curve dinamiche, un approccio differente che utilizza oracoli di prezzo di mercato per aggiornare automaticamente il rapporto tra gli asset nel pool, eliminando le opportunità di arbitraggio e garantendo liquidità costante anche in condizioni di forte variazione dei prezzi. È stato sottolineato, inoltre, come ci siano diversi fattori che possono influenzare la liquidità, i costi e gli incentivi nei pool [Adams et al., 2024, Chan, 2023, Zhu et al., 2024].
Un altro tema di rilievo è l'IL, poiché rappresenta una delle principali criticità degli AMM, influenzando direttamente la redditività e la sostenibilità della fornitura di liquidità. Di particolare interesse sono le sue implicazioni per i fornitori di liquidità e come tali effetti possano essere mitigati [Hafner and Dietl, 2024, Kim et al., 2022, 2024, LI et al., 2024, Tangri et al., 2023]. Risultano essere particolarmente interessanti gli studi che implementano strategie di *hedging* per far fronte a questo problema, seguendo modelli analoghi a quelli della finanza tradizionale come Black-Scholes-Merton e modelli a volatilità stocastica log-normale [Dewey and Newbold, 2023, Fukasawa et al., 2023, Khakhar and Chen, 2022, Lipton et al., 2024], proponendo anche la struttura di un *replicating portfolio* come in Clark [2020]. L'IL non sempre però è ritenuta il metodo ottimale per misurare le perdite, a tal fine in Milionis et al. [2022] viene introdotta una nuova metrica detta *loss-vs-rebalancing*.
In parallelo, in studi differenti, si è trattato il tema della *capital efficiency* che risulta cruciale per comprendere come distribuire il proprio capitale in modo ottimale [Abgaryan et al., 2024, Chen et al., 2023].
Un ulteriore argomento interessante è trattato in Capponi et al. [2024] in cui si analizza il ruolo della gestione *just-in-time* della liquidità, ovvero la capacità di rendere disponibili i fondi esattamente quando necessario, e il suo impatto sulla redditività dei fornitori di liquidità passivi.
È importante evidenziare l'esistenza di studi che permettono di ottenere un'analisi comparativa tra mercati tradizionali e DeFi, fondamentali per comprendere le potenzialità di quest'ultima [Drissi, 2023], introducendo anche strumenti utili per futuri studi, come i

derivati finanziari nel contesto delle criptovalute e analisi statistiche approfondite [Abraham, 2024, Chen et al., 2024, Fateh Singh et al., 2025, Karagiorgis et al., 2024, Matic et al., 2023, Tamandi, 2024, Zięba, 2019].
Questi studi sono stati cruciali per una comprensione profonda del tema e per selezionare gli strumenti utili per strutturare un modello che fosse in grado di rappresentare queste dinamiche in modo realistico, ottenendo risultati complementari a quelli già formalizzati. Questo elaborato, infatti, si inserisce all'interno della letteratura esistente approfondendo il tema dell'ottimizzazione dei liquidity pool con un approccio innovativo. Molti studi, come evidenziato in precedenza, si concentrano sulla modellizzazione matematica degli AMM, sulla gestione del rischio o sulle strategie di mitigazione dell'IL, al contrario questo lavoro mira, non solo a descrivere le dinamiche esistenti, ma ad integrare questi aspetti in un contesto di ottimizzazione che fornisca indicazioni pratiche per la progettazione e l'implementazione di meccanismi di incentivazione. Ciò tenendo conto anche delle interazioni tra liquidità, incentivi e comportamento degli LP. L'obiettivo, dunque, è quello di coniugare la teoria, ampiamente trattata, con le applicazioni, offrendo un modello che non solo analizzi le dinamiche di mercato, ma fornisca anche uno strumento utile per lo sviluppo di pool più efficienti e stabili. Rispetto alla letteratura esistente, inoltre, questo studio pone particolare attenzione alla relazione tra liquidità disponibile e struttura degli incentivi, evidenziando come diversi livelli di liquidità possano richiedere strategie di incentivazione differenti. L'approccio adottato si distingue per l'integrazione di un'analisi quantitativa, basata su dati empirici, che combina strumenti di regressione con tecniche di ottimizzazione, permettendo di determinare in modo rigoroso la configurazione ottimale di un liquidity pool. Così viene definita una funzione di liquidità all'interno del pool, in modo tale da inserirla in un modello di ottimizzazione, il cui fine, come già esplicato, è quello di stabilire i parametri ottimali in fase di setup. In questo modo, si contribuisce a un dibattito più ampio sulle migliori pratiche per garantire la sostenibilità economica della DeFi nel lungo periodo.
Dall'analisi condotta emerge, infatti, che esistono pattern ricorrenti nella distribuzione di liquidità e incentivi, i quali rispondono a precise dinamiche di mercato e influenzano il comportamento degli LP. In particolare, si osserva come il livello di liquidità disponibile condizioni direttamente la necessità di incentivi per attrarre e mantenere gli investitori, delineando così strategie ottimali che variano in base al contesto. Questa evidenza fornisce un quadro utile per gli sviluppatori di protocolli DeFi, offrendo indicazioni pratiche su come strutturare meccanismi di incentivazione efficienti e sostenibili.
La ricerca svolta può dunque fornire un'idea sulle caratteristiche da ricercare in fase di sviluppo di un liquidity pool, ponendo una base per ricerche più approfondite e mirate sul tema. Si offre, infatti, una prospettiva sulle future sfide e opportunità nel settore, evidenziando come l'evoluzione della DeFi e dei modelli di incentivazione possano influenzare la stabilità e l'adozione su larga scala di queste soluzioni. Lo studio può dunque essere utile come base per lo sviluppo di strategie innovative volte a migliorare l'efficienza dei mercati decentralizzati, contribuendo alla creazione di ecosistemi più stabili e sicuri.
In questo elaborato, dunque, si propone l'analisi dei fondamenti della blockchain, approfondendo i suoi meccanismi di consenso, le tipologie esistenti e le implicazioni economiche e finanziarie che ne derivano. Il secondo capitolo (2) è, infatti, dedicato all'introduzione della blockchain, partendo dalla sua nascita fino all'evoluzione attuale. Viene trattata,

inoltre, la classificazione delle blockchain, concentrandosi sul ruolo dei token e degli smart contract nella DeFi.
Si evidenziano successivamente le criticità legate all'implementazione di un modello analogo a quello utilizzato in finanza tradizionale, come gli order book decentralizzati, sottolineando l'esigenza di una soluzione più coerente con i principi di decentralizzazione e sicurezza che sono alla base della blockchain.
Nel capitolo 3 vengono esaminate le diverse tipologie di AMM, come quelli a somma costante, a media costante, ibridi e a liquidità concentrata, mettendone in evidenza le caratteristiche e analizzando i rischi ad essi associati. Particolare attenzione è dedicata al fenomeno dell'IL e alle strategie per mitigarne gli effetti.
Nel capitolo 4, vengono delineati gli obiettivi perseguiti e la metodologia adottata nello studio. Per comprendere al meglio il funzionamento e l'efficacia di questi sistemi, ci si concentra sui processi decisionali che guidano la progettazione di un liquidity pool, focalizzandosi anche sulla distribuzione degli incentivi.
Successivamente all'aver risolto tale modello viene presentata un'analisi dei risultati (capitolo 5), per poter comprendere effettivamente quali indicazioni si possono ottenere. Osservando anche casistiche che non sono state trattate in questo elaborato, ma che possono rappresentare uno spunto per studi futuri.
Si trattano infine le conclusioni tratte dell'analisi svolta nel capitolo 6.

Capitolo 2

Blockchain

2.1 Nascita della blockchain

La nascita ufficiale della **blockchain**, e di conseguenza delle **criptovalute**, si associa solitamente alla pubblicazione del whitepaper da parte di Satoshi Nakamoto *Bitcoin: A Peer-to-Peer Electronic Cash System* [Nakamoto, 2008] nel 2008. L'idea di creare un sistema decentralizzato, però, ha radici più lontane nel tempo.
Risale, infatti, al 1988 il *Crypto Anarchist Manifesto* [May, 1988] di Timothy C. May, in cui vengono introdotti per la prima volta i concetti di **anonimato** e **libertà individuale**. La necessità di affrontare questi principi nasce dall'evoluzione digitale avvenuta nell'ultimo decennio del ventesimo secolo; questa esigenza è soddisfatta mediante l'uso della **crittografia**, ossia un processo che trasforma un testo in chiaro leggibile in un testo cifrato illeggibile. Lo scopo è quello di nascondere informazioni sensibili affinché non possano essere accessibili a utenti non autorizzati.
Accompagnato dalla pubblicazione del *Cypherpunk's Manifesto* [Hughes, 1997], nasce così il movimento **Cypherpunk** fondato dallo stesso May, insieme a John Gilmore ed Eric Hughes. Il principio fondamentale dei Cypherpunk era che la privacy fosse un diritto inalienabile. In questo contesto, la crittografia rappresentava lo strumento più efficace per proteggerlo, garantendo la sicurezza personale nell'era digitale e sfidando i tentativi di sorveglianza e controllo da parte di enti governativi e corporativi.
Non solo l'ideologia che porterà alla creazione del **Bitcoin** ha origini più lontane, ma anche gli strumenti che verranno utilizzati.
Il sistema di **Proof-of-Work** (PoW), che verrà introdotto in modo più approfondito nel capitolo 2.3.2, infatti, è anch'esso stato precedentemente introdotto da Adam Back nel 1997 con **Hashcash** [Adams et al., 2021]. Lo scopo era quello di limitare le e-mail di spam e gli attacchi *denial of service* (DoS), il cui intento è quello di rendere indisponibili le risorse di un sistema informatico causando un'interruzione dei servizi.
Nell'anno successivo, 1998, si ha la prima idea di criptovaluta da parte di Wei Dai, che propone ai Cypherpunk **B-money** [Dai, 1998]: una moneta che avrebbe permesso pagamenti anonimi e decentralizzati grazie all'utilizzo di Hashcash. Questo progetto, però, non è mai stato effettivamente implementato.

Nel 2004, il progetto **Reusable PoW** di Hal Finney, [Finney, 2004], ha incontrato lo stesso destino, non giungendo mai a una realizzazione concreta. La valuta sarebbe stata creata sfruttando Hashcash anche in questo caso.
La proposta più affine al Bitcoin attuale arriva nel 2005 da Nick Szabo con **Bit Gold**, [Szabo, 2005].
Il fallimento di Lehman Brothers Holdings Inc. nel 2008, infine, ha evidenziato le criticità del sistema economico centralizzato, lasciando spazio allo sviluppo di monete digitali come il Bitcoin.
Il nome Satoshi Nakamoto, a cui, come precedentemente indicato, si attribuisce la pubblicazione del whitepaper di Bitcoin [Nakamoto, 2008], è uno pseudonimo utilizzato da una o più persone che hanno effettivamente contribuito alla realizzazione del progetto Bitcoin. Molti ritengono, infatti, che dietro al nome fittizio Satoshi Nakamoto possano celarsi Wei Dai, Hal Finney, Nick Szabo o altri membri del gruppo di sviluppatori.
Nasce così una forma di denaro elettronico completamente peer-to-peer che consente l'invio diretto di pagamento online da un ente all'altro senza la necessita di un'istituzione finanziaria intermediaria. Solo pochi mesi dopo, nel 2009, il codice sorgente viene pubblicato e viene creato il primo blocco della blockchain di Bitcoin.

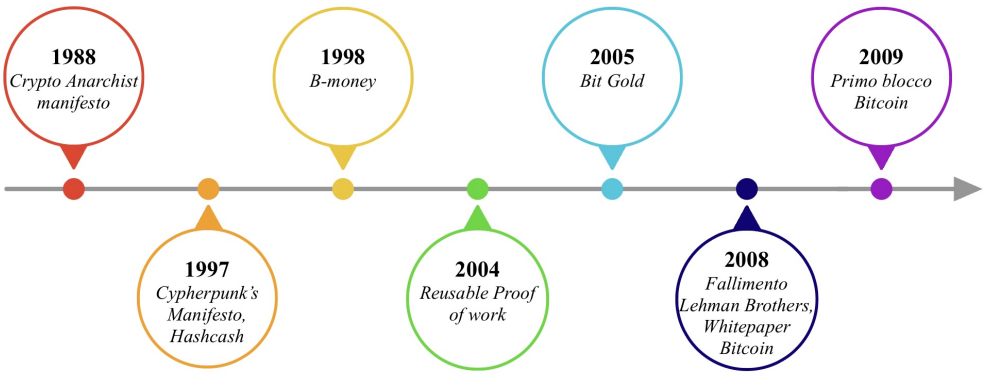


Figura 2.1. Rappresentazione degli eventi su linea del tempo

Negli anni successivi, sono state sviluppate numerose blockchain. Alcune di queste hanno avuto una vita breve, mentre altre sono diventate pilastri fondamentali nel mondo delle criptovalute e delle applicazioni decentralizzate. Tra queste ultime, spicca l'idea di Vitalik Buterin, che nel 2015 con **Ethereum** [Buterin et al., 2013], ha rivoluzionato il settore grazie all'implementazione degli smart contract, che verranno introdotti nel capitolo 2.5.

2.2 Basi della blockchain

La crittografia è il pilastro principale su cui si basa la blockchain, proprio per questo motivo è necessario introdurre alcuni concetti essenziali.
A tale scopo si definiscono le **funzioni di hash**:

Definizione: 2.2.1. *una funzione hash H è una funzione non invertibile che mappa una stringa di caratteri di lunghezza arbitraria in una stringa di lunghezza fissa detta digest.*

Le proprietà fondamentali di queste funzioni sono:

- **Resistenza alla prima preimmagine**: dato un qualsiasi digest d deve essere computazionalmente intrattabile trovare una stringa m_1 tale per cui $H(m_1) = d$;
- **Resistenza alla seconda preimmagine**: data una qualsiasi stringa m_1 deve essere computazionalmente intrattabile trovare una stringa $m_2 \neq m_1$ tale per cui $H(m_1) = H(m_2)$;
- **Resistenza alle collisioni**: deve essere computazionalmente intrattabile trovare una qualsiasi coppia di stringe $m_1 \neq m_2$ tali per cui $H(m_1) = H(m_2)$.

Ciò è importante poiché risulta impossibile cambiare la stringa di partenza senza alcuna conseguenza sull'output.
Nel corso degli anni, sono state sviluppate e implementate diverse funzioni hash con livelli di sicurezza sempre maggiori. Le nuove versioni hanno risolto le vulnerabilità emerse in quelle precedenti come, ad esempio, nel caso degli algoritmi MD5 e SHA-1, originariamente considerati sicuri, ma successivamente sostituiti poiché non resistenti alle collisioni. Attualmente si utilizzano algoritmi più sicuri che offrono una maggiore resistenza agli attacchi informatici, come SHA-256 (Secure Hash Algorithm a 256 bit), e SHA-3 (Secure Hash Algorithm 3), derivato da Keccak-256. Quest'ultimo è impiegato da Ethereum, mentre il primo è utilizzato da Bitcoin.
Un altro concetto base è quello di **crittografia asimmetrica** o **crittografia a chiave pubblica**:

Definizione: 2.2.2. *la crittografia asimmetrica è un sistema crittografico che utilizza una coppia di chiavi differenti: la chiave pubblica, nota a tutti, e la chiave privata, nota solo al suo proprietario.*

Le due chiavi sono strettamente legate perché solo la chiave privata può decifrare i dati criptati dalla chiave pubblica corrispondente e viceversa, garantendo così che le informazioni siano accessibili esclusivamente a chi possiede la chiave corretta. Gli algoritmi utilizzati nella crittografia asimmetrica, infatti, garantiscono che non sia possibile determinare la chiave privata a partire dalla chiave pubblica. Questo meccanismo di sicurezza è fondamentale per proteggere le informazioni e garantire l'autenticità e la riservatezza delle comunicazioni. Risulta quindi più sicura della crittografia simmetrica in cui si utilizza la stessa chiave per cifrare e decifrare. I vantaggi di questo meccanismo sono:

- Distribuzione delle chiavi non necessaria: per anni, i canali di distribuzione delle chiavi poco sicuri sono stati un problema significativo. Tuttavia, la crittografia asimmetrica risolve questo problema, poiché le chiavi pubbliche possono essere scambiate tramite server appositi. Se anche una di esse venisse divulgata, la sicurezza dei dati cifrati rimane intatta, poiché non è possibile derivare la chiave privata dalla chiave pubblica;

- Firma digitale o autenticazione del messaggio: grazie ai processi di crittografia asimmetrica, il mittente può utilizzare la chiave privata per firmare digitalmente il messaggio o il file, garantendo così che provenga da soggetti autenticati;

- Trasferimento non necessario delle chiavi private: la crittografia asimmetrica garantisce la sicurezza delle chiavi private, che non vengono mai trasmesse attraverso canali di comunicazione compromessi o potenzialmente compromessi.

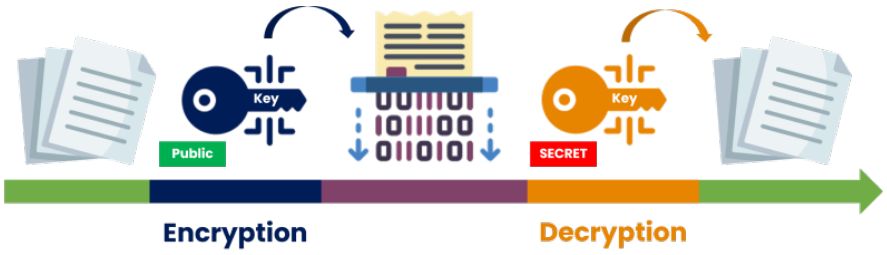


Figura 2.2. Funzionamento crittografia asimmetrica[1]

Un semplice esempio per comprendere il funzionamento della crittografia asimmetrica: Alice vuole inviare un documento a Bob e vuole essere sicura che solo lui sia in grado di leggerlo. Utilizza dunque la chiave pubblica di Bob per cifrare il documento, una volta cifrato lo invia attraverso un qualsiasi canale di comunicazione, perché solo la chiave privata di Bob può decifrare il documento. Così, anche se il documento cifrato viene intercettato, nessuno, eccetto Bob, sarebbe in grado di leggerlo.
L'importanza di queste definizioni è legata al funzionamento della blockchain per eccellenza: Bitcoin.
In questo contesto, infatti, Alice e Bob hanno come obiettivo lo scambio di un bitcoin. Entrambi sono in possesso di una chiave pubblica, detta **indirizzo**, e di una chiave privata che garantisce che il trasferimento di beni sia volontario. Per inviare un bitcoin a Bob, Alice sfrutta la crittografia asimmetrica. Cripta la transazione utilizzando la chiave pubblica di Bob, autorizza la transazione con la sua chiave privata e la invia all'indirizzo di Bob.
Un altro aspetto fondamentale legato alla sicurezza, ottenuto con gli stessi mezzi, è la firma digitale. Supponendo di prendere sempre in considerazione il caso in cui Alice manda un messaggio a Bob, i passaggi da seguire per l'utilizzo sono:

1. Alice scrive un messaggio e applica la funzione di hash ottenendo il digest;

2. Il digest viene crittografato con la chiave privata di Alice, ottenendo la firma digitale;

3. Viene inviato il messaggio originale, con firma digitale allegata;

[1]Fonte: https://www.sealpath.com/it/blog/tipi-di-crittografia-guida/.

4. Bob separa il messaggio dalla firma;

5. Bob a questo punto decripta la firma digitale, utilizzando la chiave pubblica di Alice;

6. Sempre Bob applica la funzione di hash al messaggio;

7. Verifica, infine, che il digest sia uguale all'output ottenuto al passaggio precedente.

La verifica consentirà di attestare che chiunque abbia inserito quei dati era in possesso della chiave privata corrispondente alla chiave pubblica fornita. Questo processo garantisce che la stringa di codice utilizzata come firma digitale sia stata generata dalla chiave privata associata, confermando così l'autenticità e l'integrità dei dati.
Prima di approfondire ulteriormente l'utilizzo di hash e crittografia asimmetrica, però, è necessario introdurre il concetto di **blockchain**.
La blockchain, letteralmente "catena di blocchi", offre la possibilità di registrare dati e transazioni in modo permanente e immutabile. Questa caratteristica è essenziale perché garantisce la sicurezza dello scambio di beni senza la necessità di un intermediario.
È stata introdotta per la prima volta con lo scopo di fornire un registro pubblico, noto come **ledger**, per documentare le transazioni di Bitcoin. L'obiettivo era quello di ottenere un database sicuro, condiviso attraverso una rete di partecipanti, in cui le informazioni aggiornate sono accessibili contemporaneamente a tutti.
La tecnologia blockchain appartiene, infatti, alle cosiddette **DLT** (Distributed Ledger Technologies), che si basano sul concetto di un registro distribuito, e ne rappresenta sola una delle molteplici forme.
Per rendere questo concetto più chiaro, si possono citare le definizioni fornite dalla Banca Centrale e dalla World Bank. La prima descrive la DLT come una tecnologia che "permette agli utenti di memorizzare e accedere alle informazioni relative a un dato insieme di attività. Queste informazioni sono distribuite tra gli utenti, che potrebbero poi usarle per regolare i loro trasferimenti, ad esempio di titoli e contanti, senza dover fare affidamento su un sistema di convalida centrale di fiducia" [Pinna and Ruttenberg, 2016]. La seconda definisce i sistemi DLT come "una tecnologia che consente di registrare, condividere e sincronizzare transazioni e dati in una rete distribuita con diversi partecipanti" [Natarajan et al., 2017].
Queste definizioni aiutano a comprendere meglio il concetto di DLT e la sua applicazione pratica nel contesto della blockchain. Si tratta di un cambiamento epocale. In precedenza, esistevano solo i cosiddetti *Central Ledger*, registri digitali centralizzati utilizzati per documentare le transazioni e la proprietà di beni o attività. Questi registri erano gestiti da un'entità centrale, come una banca o un'altra autorità di regolamentazione, e servivano a memorizzare transazioni in valuta, titoli, proprietà immobiliari e altri beni.
In generale, un Central Ledger è un sistema di registrazione centralizzato, dove tutte le transazioni sono registrate in un unico registro condiviso, invece di registri distribuiti come in un sistema decentralizzato come per le Blockchain.
In un Distributed Ledger, invece, si ha un registro che si aggiorna automaticamente e identicamente su ciascuno dei nodi che partecipano alla rete, grazie a diversi meccanismi di governance. Si può notare meglio la differenza con lo schema in figura 2.3.

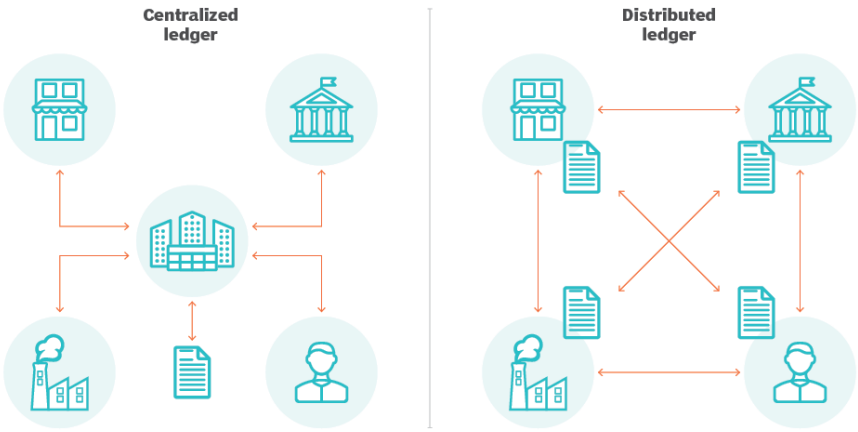


Figura 2.3. Differenza tra centralized e distributed ledger[2]

Nel caso della blockchain ogni nodo può contribuire alla validazione delle transazioni, così che possano essere inserite in un nuovo blocco. Questo processo di validazione è reso possibile grazie all'uso di specifici **algoritmi di consenso**, che garantiscono l'integrità e la sicurezza del registro distribuito e verranno introdotti dettagliatamente nella sezione successiva 2.3.

2.3 Algoritmi di consenso

Considerando la struttura dei DLT descritta in 2.2 è fondamentale costruire un sistema che permetta a tutti i partecipanti del network di fidarsi reciprocamente, in modo da poter validare le transazioni e aggiungerle alla catena. Questo meccanismo viene introdotto per evitare l'inserimento di transazioni scorrette o fraudolente.
Prima di comprendere come avviene l'inserimento delle transazioni è importante conoscere la struttura di un blocco. Si prenderà in esempio la struttura della blockchain di Bitcoin per comprendere meglio il funzionamento.
Un blocco è composto da due elementi chiave, ossia l'intestazione del blocco (**block header**) e il **corpo del blocco**. Il primo contiene:

- Versione del blocco: indica quale versione del protocollo è in uso.
- Hash del blocco precedente: hash crittografico del blocco precedente nella catena.
- Merkle Root: hash radice del Merkle Tree, che riassume tutte le transazioni nel blocco.

[2]Fonte: https://acowebs.com/blockchain-in-ecommerce/.

- Timestamp: la data e l'ora in cui il blocco è stato creato.
- Difficoltà: il livello di difficoltà del puzzle di mining che il nodo ha risolto per creare il blocco.
- Nonce: un numero casuale di 32 bit utilizzato dal miner per ottenere un blocco valido.

Versione	02000000
Hash del blocco precedente (PrevHash)	E87C17C45768w7e1643fsd5481sd3f4131df681
Merkle root	697we168t4v1a4rv3v1e3r43c4er14ca8c4168a
Timestamp	358b0553
Bits	535f0119
Nonce	48750933

Tabella 2.1. Esempio struttura dell'header di un blocco Bitcoin

Il secondo, invece, contiene l'elenco di tutte le transazioni.
Conoscendo questa struttura, è possibile comprendere meglio come avviene il processo di validazione e inserimento delle transazioni nella blockchain.
Sono tutte informazioni fondamentali per verificare l'integrità della chain. La **Merkle root**, ad esempio, è fondamentale per verificare rapidamente e in modo efficiente l'integrità e l'autenticità delle transazioni contenute nel blocco stesso. Essa rappresenta, infatti, un riassunto di tutte le transazioni all'interno del blocco, contenute nel **Merkle Tree**, o hash tree. Una qualsiasi modifica effettuata anche ad una singola transazione cambierebbe drasticamente il valore di questo campo, rendendo così impossibile l'alterazione dei dati. Uno schema esemplificativo della struttura del Merkle Tree è riportato in figura 2.4.
Queste informazioni permettono di avere un quadro completo per analizzare i diversi algoritmi di consenso utilizzati per garantire l'integrità e la sicurezza del registro distribuito.
Come suggerisce il nome, un algoritmo di consenso è un insieme di istruzioni progettato per risolvere un problema. Nel contesto della blockchain, l'obiettivo è garantire un ambiente privo di errori in cui tutte le copie del database siano allineate sulle informazioni contenute. Conflitti e dati discordanti non sono accettabili, poiché renderebbero la catena praticamente inutile.
Oltre al meccanismo di PoW, presentato in precedenza, esistono diversi algoritmi di consenso, come:

- Proof of Stake
- Delegated Proof of Stake
- Pure Proof of Stake

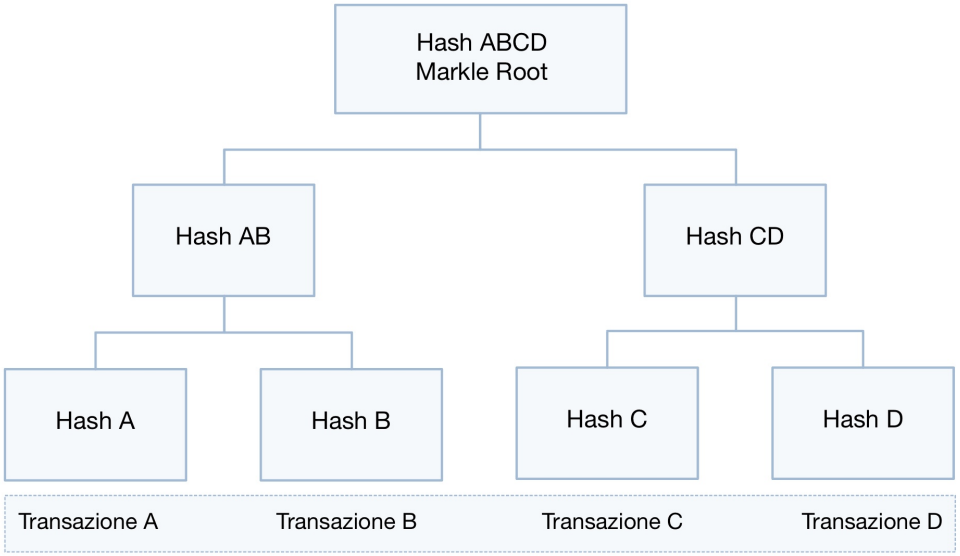


Figura 2.4. Merkle Tree

- Proof of Importance
- Proof of Authority
- Proof of Burn
- Proof of Activity
- Proof of Space

Tra i precedenti algoritmi citati, verranno introdotti solo quelli maggiormente diffusi. Si trattano in dettaglio, quindi, la PoW, 2.3.2, e la Proof of Stake (PoS), 2.3.3, per fornire una comprensione completa dei meccanismi che governano le reti blockchain più popolari. La principale sfida che gli algoritmi di consenso devono superare è il "**trilemma della blockchain**" (figura 2.5). Questo problema è stato formulato da Vitalik Buterin, il creatore di Ethereum. Afferma che è quasi impossibile sviluppare una blockchain che soddisfi contemporaneamente queste tre proprietà fondamentali: scalabilità, sicurezza e decentralizzazione.
Il miglioramento di una di queste caratteristiche comporta inevitabilmente un compromesso a discapito di almeno una delle altre. I principi che devono essere rispettati sono:

- **Decentralizzazione**: l'intera struttura è progettata in modo che nessuno detenga il comando assoluto, il controllo è completamente distribuito in modo che tutti i partecipanti abbiano accesso agli stessi dati;

Figura 2.5. Trilemma della Blockchain

- **Sicurezza**: una rete blockchain efficace deve essere resistente agli attacchi e garantire l'integrità e la fiducia dei dati;
- **Scalabilità**: una buona rete blockchain deve essere in grado di espandersi e adattarsi alla crescita della domanda, mantenendo tempi di transazione rapidi e costi di gestione accessibili.

L'obiettivo è dunque quello di trovare un equilibrio tra scalabilità, sicurezza e decentralizzazione per garantire un'esperienza ottimale anche con un alto volume di attività sulla rete. Un incremento della scalabilità, ad esempio, potrebbe compromettere la decentralizzazione, poiché potrebbe richiedere l'adozione di nodi più potenti e meno distribuiti, favorendo la concentrazione del controllo. Analogamente, privilegiare la sicurezza può comportare una riduzione dell'efficienza e della scalabilità del sistema. Questo equilibrio complesso è noto come il trilemma della blockchain.

2.3.1 Byzantine Fault Tolerance

Per comprendere l'utilità degli algoritmi di consenso può essere funzionale introdurre il problema matematico noto come "Problema dei generali bizantini", o "**The Byzantine Generals Problem**", discusso per la prima volta nel 1982 da tre matematici: Marshall Pease, Robert Shostak e Leslie Lamport [Lamport et al., 2019].
La metafora utilizzata vuole rappresentare la capacità di raggiungere un consenso anche in presenza di informazioni imperfette. Il Problema dei Generali Bizantini è, infatti, un problema di teoria dei giochi che rivela le difficoltà di raggiungere il consenso tra un gruppo di entità reciprocamente sospettose che utilizzano canali di comunicazione inaffidabili (figura 2.6). Questo problema evidenzia le sfide nel coordinamento e nella fiducia necessari per prendere decisioni collettive in ambienti incerti e potenzialmente ostili. La teoria dei

giochi si riferisce alla migliore strategia adottata da attori indipendenti e in competizione nel processo decisionale.
In questo esempio si ha un gruppo di generali che assedia una città e deve decidere se attaccare o ritirarsi, a tal fine possono scambiarsi informazioni. Perché l'attacco vada a buon fine devono essere tutti concordi, se anche uno solo dei generali non è d'accordo, l'attacco rischia di fallire.
Lo scopo è quindi quello di raggiungere un consenso unanime, nonostante esista la possibilità di trovare alcuni generali disonesti o che i messaggi possano essere corrotti durante il trasporto.
Le soluzioni trovate da Pease, Shostak e Lamport sono diverse, ma quella di maggiore rilevanza è associata ad un algoritmo detto **Oral Messages** (OM), ottenuto risolvendo il problema "commander and lieutenants". In questo scenario il comandante invia ordini ai luogotenenti, i quali possono scambiarsi messaggi tra loro. Tutti i protagonisti di questo problema possono essere traditori, ma l'obiettivo è comunque raggiungere un consenso in maggioranza.
Le assunzioni fatte per poter applicare questo algoritmo sono:

- una volta che il messaggio è stato inviato arriverà al destinatario;
- il destinatario è in grado di risalire esattamente a chi lo ha inviato;
- se qualcuno non inviasse il messaggio gli altri ne sarebbero a conoscenza.

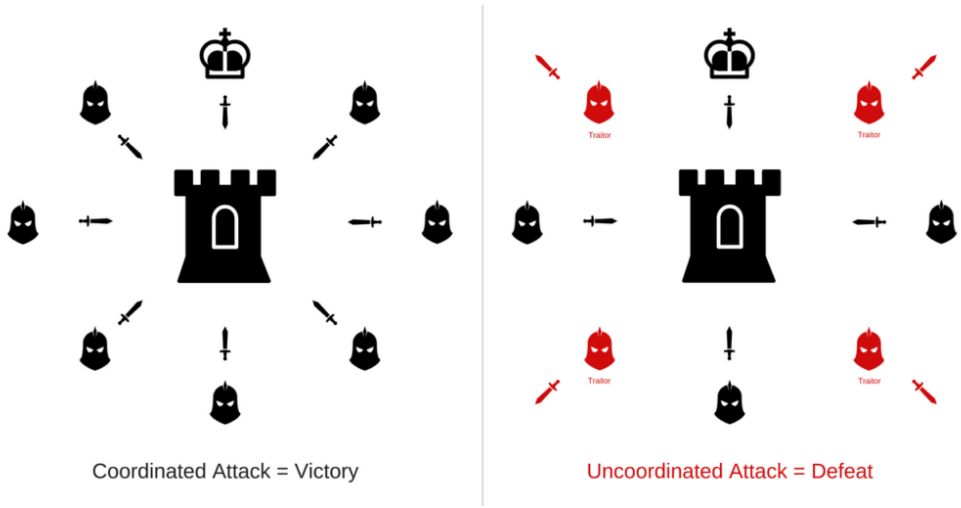


Figura 2.6. Problema dei generali bizantini[3]

Da ciò deriva il seguente teorema:

[3]Fonte: https://medium.com/coinmonks/the-byzantine-generals-problem-d6c9d2b2eb9.

Teorema: 2.3.1. *sia m il numero di traditori, l'algoritmo OM(m) può raggiungere il consenso se ci sono più di 3m generali totali.*

Più semplicemente, il consenso può essere raggiunto quando le persone leali saranno almeno $\frac{2}{3}$. Si può dimostrare che sapendo risolvere questo sottoproblema si risolve anche il problema dei generali bizantini originale.
È evidente l'analogia con la blockchain, in cui molte persone devono accordarsi per raggiungere il consenso della rete, senza sapere se alcune di esse stiano agendo in modo malevolo. In questo contesto, se i nodi disonesti riescono a controllare il 51% del network, la blockchain è compromessa.
Da queste osservazioni prende vita la definizione di **Byzantine Fault Tolerance** (BFT), ovvero la resistenza e l'integrità di una chain in presenza di nodi malevoli.
Nel 1999 è stato introdotto da due informatici un algoritmo noto come **Practical Byzantine Fault Tolerance** (PBFT) [Castro et al., 1999], in grado di gestire migliaia di richieste al secondo con bassissima latenza.
Sono molti gli algoritmi di questa famiglia che sono stati sviluppati, un esempio è l'*efficient synchronous byzantine consensus* [Abraham et al., 2017], e vengono utilizzati anche in contesti diversi dalla blockchain, come nei sistemi di controllo di motori di aerei o missili, che ricevono segnali da sensori differenti e che a volte possono differire. L'applicazione di tali algoritmi in questi ambiti garantisce che, nonostante possibili guasti o malfunzionamenti di alcuni sensori, il sistema nel suo complesso possa comunque raggiungere un consenso affidabile e funzionare correttamente. Pur essendoci svariati algoritmi di consenso, tutti mirano ad assicurare il corretto funzionamento del network anche in queste condizioni.

2.3.2 Proof of Work

La **PoW** è l'algoritmo di consenso più famoso, in quanto quello utilizzato nella rete Bitcoin e il primo ad essere stato implementato. Non fa parte degli algoritmi derivanti direttamente dai BFT, ma riesce comunque a conferire alla rete le stesse caratteristiche.
Questo sistema prevede che per l'inserimento di nuovi blocchi si fornisca, come suggerisce il nome, una prova del lavoro compiuto.
In particolare, per la validazione è necessario risolvere una challenge che consiste nel trovare un hash delle transazioni, concatenato al nonce, che sia inferiore a un determinato valore target. Come spiegato in precedenza, sia il valore del nonce che quello del target sono riportati nell'header del blocco.
Proprio a causa della complessità dei calcoli richiesti, sono stati progettati strumenti specifici, chiamati ASIC (Application-Specific Integrated Circuits), in grado di eseguire questo lavoro in modo più efficiente e rapido.
Chi è in grado di risolvere la sfida più velocemente ha la possibilità di creare il blocco e, come ricompensa, riceve una quantità di criptovaluta appena creata. Proprio per questo motivo, il processo viene chiamato *mining* poiché i "minatori" di criptovaluta devono risolvere complessi problemi matematici per scoprire nuovi blocchi e guadagnare ricompense.
Tornando all'analogia dei generali bizantini, possiamo immaginare che ogni decisione presa dai generali sia accompagnata da un problema complesso da risolvere, la cui soluzione

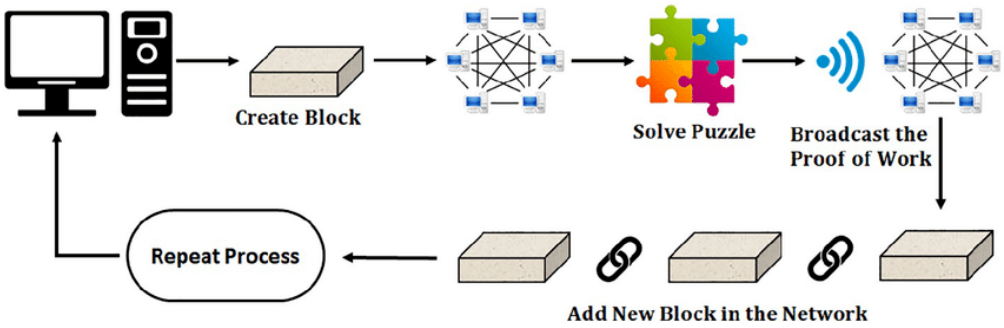


Figura 2.7. Processo seguito dalla PoW [Abukari et al., 2023]

deve essere inclusa nel messaggio per essere verificata dagli altri generali. In questo scenario, un generale traditore non potrebbe inviare messaggi discordanti poiché non avrebbe abbastanza tempo per risolvere tutti i problemi necessari.
Questo algoritmo è resistente al cosiddetto 51% attack. Per manipolare le transazioni, un miner dovrebbe possedere il 51% della potenza di calcolo della rete, il che diventa sempre più difficile da ottenere via via che il numero di nodi del network aumenta. Quando la rete raggiunge livelli di estensione elevati, l'attività di manipolazione diventa virtualmente impossibile. Questa caratteristica rende la PoW estremamente sicura e affidabile per la validazione delle transazioni e il mantenimento dell'integrità della blockchain.
La difficoltà della PoW viene regolata automaticamente per garantire che il tempo medio tra la validazione di un blocco e quello successivo rimanga costante, tipicamente ogni 10 minuti (come nel caso del Bitcoin). Se il tempo impiegato per validare un blocco è significativamente superiore, la difficoltà viene ridotta; viceversa, se il tempo necessario per il mining è troppo breve, la difficoltà viene aumentata. Questo meccanismo assicura che la rete mantenga un ritmo costante e prevedibile nella creazione di nuovi blocchi.
L'attività di mining richiede l'utilizzo dispositivi molto costosi, che consumano anche grandi quantità di energia elettrica, aumentando ulteriormente i costi. Questo fenomeno rappresenta una minaccia alla decentralizzazione del sistema in quanto solo una piccola parte degli utenti può permettersi di fare tali investimenti. Di conseguenza, dal 2009 ad oggi, sono stati sviluppati altri algoritmi di consenso, come la PoS, al fine di superare queste criticità presenti.

2.3.3 Proof of Stake

La PoS è un algoritmo di consenso implementato per la prima volta sulla rete Ethereum. Si è diffusa rapidamente grazie alla sua maggiore efficienza energetica e scalabilità rispetto alla PoW.
In questo algoritmo il ruolo dei miner viene sostituito dai **validatori**, i quali sono responsabili della creazione di nuovi blocchi e della verifica delle transazioni.
I validatori vengono scelti in base alla quantità di criptovaluta che possiedono e decidono di mettere in *staking*. Questo meccanismo consiste nel mettere una parte di cripto valuta

come garanzia, dimostrando così il loro impegno e interesse a mantenere la sicurezza e l'integrità della rete. Se dovessero agire in modo malevolo o scorretto, rischierebbero di perdere la criptovaluta messa in staking, incentivando così comportamenti onesti e corretti.
I validatori vengono sorteggiati, la probabilità di essere scelti solitamente è proporzionale alla quantità di valuta messa in staking. Esistono, però, diverse evoluzioni di questo algoritmo che considerano anche altri fattori, come ad esempio il tempo da cui un validatore è presente sulla rete. Con questo approccio si mira a bilanciare il sistema, premiando non solo la quantità di valuta messa in gioco, ma anche la longevità e l'affidabilità dei validatori, rendendo la rete più robusta e sicura.
Una variante rilevante è la **Delegated PoS** (DPoS), in cui i possessori di token votano per un numero limitato di delegati che saranno responsabili della convalida delle transazioni e della creazione di nuovi blocchi. Questo sistema combina la decentralizzazione con una maggiore efficienza nella gestione della rete. La DPoS è considerata uno dei meccanismi di consenso più democratici perché offre a tutti i detentori del token la possibilità di partecipare alla creazione dei blocchi e ricevere ricompense, non solo ai grandi detentori [Gervais et al., 2016]. Questo sistema è in grado di rispondere alla principale critica rivolta alla PoS, ovvero che tende a favorire i grandi detentori di criptovalute. Essi, infatti, vengono selezionati più frequentemente per validare i blocchi e ottenere incentivi. Questo può portare a una concentrazione del potere nelle mani di pochi, minando il principio di decentralizzazione che sta alla base delle criptovalute.

Figura 2.8. Processo seguito dalla PoS[4]

[4]Fonte: https://affidaty.io/blog/it/2019/08/blockchain-la-proof-of-stake-pos/.

La sicurezza della PoS deriva dal numero di nodi che partecipano alla rete; più nodi partecipano, più sicura diventa la blockchain. In questo caso se un utente possiede più del 50% delle criptovalute messe in staking, allora è in grado di annullare il consenso e modificare i dati della catena per trarne vantaggio. Sebbene sia un'eventualità rara, rappresenta un rischio teorico che sottolinea l'importanza di una distribuzione equilibrata delle criptovalute tra i partecipanti alla rete per mantenere la sicurezza e l'integrità del sistema.
Un validatore, se scelto per verificare le transazioni di un blocco, riceverà in cambio le commissioni, o *fee*, associate a quelle transazioni. Le commissioni di transazione agiscono quindi come un incentivo economico, garantendo che il processo di validazione rimanga affidabile e continuo.
Questo sistema non solo riduce il consumo energetico, rispetto alla PoW, ma migliora anche la velocità e la capacità di gestione delle transazioni sulla rete.

2.4 Tipologie di blockchain

Le blockchain si possono classificare in categorie in base alle loro caratteristiche, al fine di comprendere l'evoluzione delle fasi di sviluppo. Due esperti in materia come Melanie Swan, [Swan, 2015], e Don Tapscott, [Tapscott and Tapscott, 2016], hanno individuato tre gruppi per distinguere i vari progetti nati su un ecosistema blockchain. Questa suddivisione si basa principalmente sulle differenti generazioni della tecnologia blockchain:

- **Blockchain 1.0**: blockchain di valore.
- **Blockchain 2.0**: blockchain di contratti intelligenti.
- **Blockchain 3.0**: blockchain scalabili e interconnesse.

La prima tipologia, infatti, coincide con la prima generazione delle tecnologie blockchain. La rete più importante associata a questa categoria è rappresentata da Bitcoin.
È evidente che il principale obiettivo sia quello di creare una valuta decentralizzata per permettere transazioni peer-to-peer, ciò concentrandosi anche sulla sicurezza e l'immutabilità dei registri di transazioni. Si ottiene così un sistema di trasferimento di denaro elettronico automatizzato, in grado di gestire le transazioni senza gravi problemi, come quello della doppia spesa. L'algoritmo di consenso utilizzato maggiormente per raggiungere questo scopo è la PoW.
La seconda categoria presenta un'evoluzione importante rispetto alla prima. Con la tecnologia di seconda generazione si è in grado di introdurre i contratti intelligenti, spiegati dettagliatamente nel paragrafo 2.5. L'esempio principale è Ethereum, che introduce per la prima volta questo tipo di possibilità.
L'esecuzione automatica di questi contratti permette di introdurre transazioni programmabili, consentendo la creazione di applicazioni decentralizzate (DApp) che possono eseguire una vasta gamma di operazioni automatizzate.
L'attività principale della blockchain, dunque, non è più esclusivamente lo scambio di denaro. Si apre la possibilità di creare applicazioni più complesse, per lo scambio di qualsiasi tipo di asset.
La blockchain 3.0, infine, rappresenta la categoria appartenente alla terza generazione

della tecnologia. L'obiettivo è quello di agire sulle limitazioni delle versioni precedenti per ottenere un ecosistema maggiormente scalabile e interconnesso. Sarebbe così possibile svolgere un numero maggiore di transazioni al secondo, consentendo anche la comunicazione e lo scambio di dati tra diverse blockchain.
Questa generazione mira, quindi, a rendere le blockchain più adatte all'adozione su larga scala e all'integrazione con altre tecnologie e sistemi [Maesa and Mori, 2020]. Esempi di reti appartenenti a questa categoria sono Ripple, Cardano e EOS.
Quello appena descritto non è l'unico metodo utilizzato per classificare le blockchain. Esse, infatti, possono essere classificate in base al livello di accessibilità della rete. Si ha così una distinzione in base a chi ha il diritto di partecipare al network e in che modo può farlo. Anche in questo caso si dividono in tre distinte categorie:

- **Public blockchains**: sono blockchain pubbliche accessibili a tutti e permettono la partecipazione libera di individui e organizzazioni senza restrizioni. Chiunque può diventare un partecipante, validare le transazioni e contribuire alla sicurezza della rete. Un esempio noto di blockchain pubblica è Bitcoin, dove ogni persona può unirsi alla rete, inviare e ricevere transazioni e partecipare al processo di verifica tramite il mining. Le blockchain pubbliche sono decentralizzate e si basano su protocolli di consenso che assicurano l'integrità e la sicurezza dei dati.

- **Consortium blockchains**: sono note anche come blockchain ibride poiché sono gestite da un consorzio di organizzazioni o entità preselezionate. I partecipanti, infatti, sono limitati a un gruppo ristretto e autorizzato di entità. Queste reti sono progettate per consentire la condivisione di dati e transazioni tra le organizzazioni coinvolte, garantendo un certo grado di controllo e privacy. La governance e la gestione della blockchain sono generalmente definite da accordi tra i membri del consorzio. Un esempio di questo tipo è R3 Corda, utilizzata in settori come la finanza e la logistica.

- **Fully private blockchains**: sono blockchain completamente private, gestite e controllate da un'unica entità o organizzazione specifica. L'accesso e la partecipazione sono limitati, infatti, esclusivamente ai membri autorizzati dall'entità che controlla la rete. Questo significa che le transazioni e i dati presenti sulla blockchain sono accessibili solo agli utenti approvati all'interno di quella particolare organizzazione. Sono utilizzate dunque per applicazioni interne, come la gestione delle forniture o la condivisione di dati sensibili tra diversi dipartimenti. Un esempio di blockchain completamente privata è Quorum, sviluppata da JPMorgan Chase.

In sintesi, le blockchain pubbliche sono accessibili a chiunque, le blockchain consortili sono limitate a un gruppo selezionato di entità, e le blockchain completamente private sono gestite e controllate da un'entità specifica. La scelta tra queste tipologie di blockchain dipende dai requisiti di accesso, controllo, privacy e sicurezza della rete blockchain per uno specifico caso d'uso.
Scegliere il giusto ecosistema blockchain per lo sviluppo di una rete risulta importante per ottenere il risultato atteso.

2.5 Token e Smart Contract

Lo sviluppo della blockchain ha rivoluzionato anche la percezione dello scambio di beni, slegandolo dai principi economici tradizionali. Questo cambiamento è avvenuto grazie all'introduzione dei **token**, che rappresentano una nuova forma di valore e permettono lo scambio all'interno delle reti decentralizzate. Vengono utilizzati per rappresentare vari tipi di valore, come asset digitali, diritti di voto, e accesso a servizi specifici.
Solitamente con il termine token ci si riferisce ad ogni tipo di criptoasset, fatta eccezione per i Bitcoin e talvolta per gli Ether, criptovaluta del sistema Ethereum. Per questo motivo può essere utilizzato per riferirsi ad un gruppo di informazioni digitali inserite all'interno di una blockchain. Conferendo, così, a un dato soggetto un determinato diritto che può riguardare la proprietà di un asset, l'accesso a un bene o a un servizio, la realizzazione o la ricezione di un pagamento. Il tutto essendo registrato all'interno di una blockchain fa sì che gli scambi di token avvengano in maniera sicura e senza intermediari [Shin, 2019].
Un semplice esempio per pensare ai token è immaginarli come le tessere magnetiche di un hotel. Solitamente una persona, dopo che la sua identità è stata verificata e la prenotazione confermata, riceve una chiave magnetica che può essere utilizzata per accedere a diversi servizi. In questo modo si evitano di ripetere tutte le azioni di verifica necessarie.
I token possono dunque essere visti come l'*incapsulamento* di valore in unità di conto scambiabili e negoziabili, come una forma di digitalizzazione del valore [Freni et al., 2022].
La "tokenizzazione" non si limita solo agli aspetti economici, ma si estende anche ad altre aree come la reputazione, il lavoro, il diritto d'autore e il diritto di voto. Una volta che questi valori vengono registrati, possono essere scambiati, contabilizzati e utilizzati all'interno di un sistema di incentivi che promuove livelli equi di ricchezza e disintermediazione. Questo permette la realizzazione concreta di una **sharing economy**, eliminando la concentrazione del potere nelle mani di pochi intermediari.
Il valore e lo scopo dei token sono determinati da chi li crea e dal contesto in cui operano. Se le criptovalute possono essere paragonate alle valute tradizionali, essendo utilizzate come mezzo di scambio, i token hanno molte altre applicazioni oltre allo scambio di valuta digitale.
Tra i principali tipi di token si hanno:

- **Exchange Token**: includono criptovalute utilizzate per effettuare pagamenti per l'acquisto di beni o servizi. Questi token sono basati su una piattaforma DLT e non sono né emessi né controllati da una banca centrale. Non forniscono alcun diritto o accesso a servizi specifici, ma sono utilizzati come strumenti di scambio o per scopi di investimento. Esempi rappresentativi di questa categoria di token includono Bitcoin, Ether e Monero.

- **Utility Token**: sono utilizzati per consentire l'accesso a prodotti o servizi offerti tramite una piattaforma DLT. Non possono essere considerati valori mobiliari e conferiscono solo il diritto di accesso a un'utilità o a un servizio digitale, che deve essere esercitabile dal momento dell'emissione del token. Se uno utility token ha anche una funzione di investimento, viene classificato come security token. Un esempio

di utility token è MANA, basato su Ethereum, utilizzato nella piattaforma di gioco virtuale Decentraland per acquistare terreni, beni e servizi.

- **Security Token**: rappresentano la proprietà di un asset digitale o fisico e hanno un valore collegato ad esso. Consentono ai proprietari di monitorare le transazioni relative all'asset tramite la registrazione su una blockchain. A differenza degli utility token, i security token sono assimilabili a titoli finanziari e sono soggetti alle relative regolamentazioni. Sono considerati strumenti di investimento e possono generare profitti per gli acquirenti. All'interno di questa categoria, esiste una sottocategoria chiamata equity token, che consente agli acquirenti di partecipare alle decisioni dell'azienda emittente, similmente all'emissione tradizionale di azioni e soggetta a specifiche normative. I proprietari di equity token possono avere il diritto di ricevere dividendi o di esercitare il potere di voto all'interno dell'azienda.

- **Governance Token**: sono utilizzati per consentire ai titolari di partecipare alle decisioni di governance di una piattaforma blockchain. I possessori di token di governance possono votare su proposte che riguardano modifiche al protocollo, l'introduzione di nuove funzionalità o l'allocazione di risorse all'interno dell'ecosistema. La distribuzione dei governance token solitamente avviene attraverso meccanismi di distribuzione iniziale, staking o come ricompensa per la partecipazione attiva alla comunità. Un esempio noto di governance token è COMP, utilizzato sulla piattaforma Compound per permettere agli utenti di votare sulle modifiche al protocollo e sulla gestione del progetto. Permettono di promuovere una gestione decentralizzata e democratica delle piattaforme blockchain, conferendo ai partecipanti un ruolo attivo nelle decisioni che influenzano l'ecosistema.

Il mezzo fondamentale per integrare i token nelle blockchain esistenti è rappresentato dagli **smart contract**. Questi ultimi sono devi veri e propri software che automatizzano l'esecuzione di accordi, garantendo che tutte le condizioni siano rispettate senza necessità di intermediari, come rappresentato nell'immagine 2.9.
L'idea di smart contract, letteralmente contratto intelligente, nasce in modo completamente indipendente dal concetto di blockchain. Sono stati introdotti, infatti, per la prima volta da Nick Szabo [Szabo, 1996]. Gli obiettivi principali della loro implementazione sono la sicurezza e la velocità di esecuzione, al fine di ridurre i tempi associati alla contrattualistica tradizionale e abbatterne i costi.
Per introdurre questo concetto, Szabo utilizza l'esempio dei distributori automatici: un semplice sistema formato da hardware e software, che in seguito al ricevimento di una moneta e alla selezione di un prodotto è in grado di erogarlo automaticamente. Il distributore rappresenta il contratto con il fornitore e opera seguendo regole predefinite, proprio come gli smart contract.
Questi software, dunque, proprio come qualsiasi altro contratto, regolano i termini e le condizioni di un accordo tra le parti. Permettono agli sviluppatori di creare applicazioni che sfruttano la sicurezza, l'affidabilità e l'accessibilità della blockchain, aprendo le porte a sofisticate funzionalità peer-to-peer. Queste funzionalità includono servizi di richiesta e concessione prestiti e assicurativi, logistica e giochi.
Gli smart contract hanno raggiunto il loro successo grazie all'implementazione sulla rete

Ethereum, che ne ha reso possibile l'adozione su larga scala, offrendo un ambiente sicuro e decentralizzato per la loro esecuzione. I programmi, infatti, vengono messi in esecuzione su nodi validatori della blockchain e il risultato, solitamente associato ad un cambio di stato della catena, rappresenta una transazione sulla quale è necessario trovare un consenso.
Questo amplia il concetto di base su cui si fonda Bitcoin, ossia l'invio e il ricevimento di denaro senza la necessità di un intermediario di fiducia, rendendo possibile l'automazione e la decentralizzazione sicura di praticamente ogni tipo di transazione, indipendentemente dal livello di complessità.

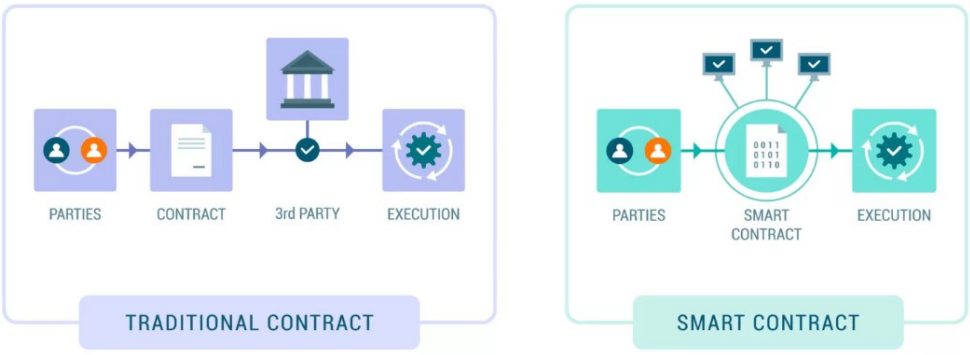


Figura 2.9. Differenza tra contratti tradizionali e smart contract [5]

Gli smart contract consentono agli sviluppatori di creare un'ampia varietà di token decentralizzati e di applicazioni. Queste ultime vengono denominate dApps (*decentralized apps*, o applicazioni decentralizzate) e una volta che vengono aggiunte alla blockchain, generalmente non possono essere cancellate o modificate.
Attualmente, Ethereum è la piattaforma di smart contract più diffusa, ma molte altre blockchain (tra cui EOS, Neo, Tezos, Tron, Polkadot e Algorand) sono in grado di eseguire questi codici. A tal fine esistono diversi linguaggi di programmazione che permettono la scrittura di questi programmi come Solidity, Web Assembly e Michelson.
Una volta che il codice è caricato sulla blockchain qualsiasi partecipante è in grado di esaminarlo, accertandosi del suo stato attuale, al fine di verificarne la funzionalità. Ogni nodo, infatti, memorizza una copia di tutti gli smart contract esistenti e del loro stato attuale.
Vengono eseguiti da tutti i nodi della rete, solo quando ricevono fondi da un utente, al fine di raggiungere il consenso in merito all'esito e al flusso di valore risultante, come anticipato. Questo meccanismo consente l'esecuzione sicura degli smart contract, senza che sia necessaria un'autorità centrale, anche nei casi in cui gli utenti si impegnano in transazioni complesse, con entità sconosciute.

[5] Fonte: https://legalfordigital.it/nft/smart-contracts-blockchain/.

Sulla rete Ethereum per eseguire i contratti intelligenti, è necessario versare una commissione denominata *gas*, consentendo alla blockchain di funzionare e garantendo che le transazioni vengano elaborate correttamente e in modo sicuro. Gli smart contract, dunque, non essendo modificabili, garantiscono che ogni processo sia a prova di censura e che non venga interrotto.
Lo sviluppo delle dApp ha permesso di includere programmi per la DeFi che puntano a trasformare il settore bancario. Le app DeFi consentono ai titolari di criptovaluta di portare a termine transazioni finanziarie complesse, gestendo risparmi, prestiti, polizze assicurative ovunque nel mondo, senza dover versare commissioni a una banca o a un istituto finanziario. Questo concetto verrà approfondito nel paragrafo 2.6.

2.6 Finanza decentralizzata

Le monete digitali nate con le blockchain e lo sviluppo degli smart contract hanno permesso lo sviluppo di un sistema finanziario ad esse legato noto come DeFi.
L'obiettivo, anche in questo caso, è quello di eliminare gli intermediari tipici della finanza tradizionale, come banche e istituzioni finanziarie. Gli utenti sfruttando la tecnologia blockchain, infatti, sono in grado di gestire transazioni e operazioni in totale sicurezza e autonomia.
Nel sistema finanziario tradizionale scambiare valute è costoso e richiede molto tempo, gli smart contract permettono invece di risolvere questo problema, rendendo più sicuro anche il prestito di liquidità ad estranei.
Si ottiene così un sistema con numerosi vantaggi:

- **Accessibilità**: chiunque con una connessione internet può accedere a servizi finanziari.
- **Decentralizzazione**: elimina la necessità di intermediari, riducendo costi e aumentando l'efficienza delle transazioni.
- **Trasparenza**: le transazioni sono pubblicamente verificabili grazie alla tecnologia blockchain, aumentando la fiducia e la sicurezza.
- **Sicurezza**: utilizza la crittografia avanzata per proteggere i dati e garantire transazioni sicure.

Queste caratteristiche hanno consentito alla DeFi di espandersi, conquistando un numero sempre maggiore di utenti. Il grafico seguente (figura 2.10), proveniente da Dune Analytics[6], evidenzia chiaramente questa crescita. Questo incremento ha portato a un conseguente aumento della capitalizzazione delle piattaforme DeFi, con un incremento di 88,5 miliardi di dollari in un solo anno, passando da 1,5 miliardi di dollari nel 2020 a 90 miliardi di dollari nel 2021, come mostrano i dati di Defi Pulse[7], nella figura 2.11. I servizi

[6]Fonte: https://dune.com/.

[7]Fonte: https://defipulse.com/.

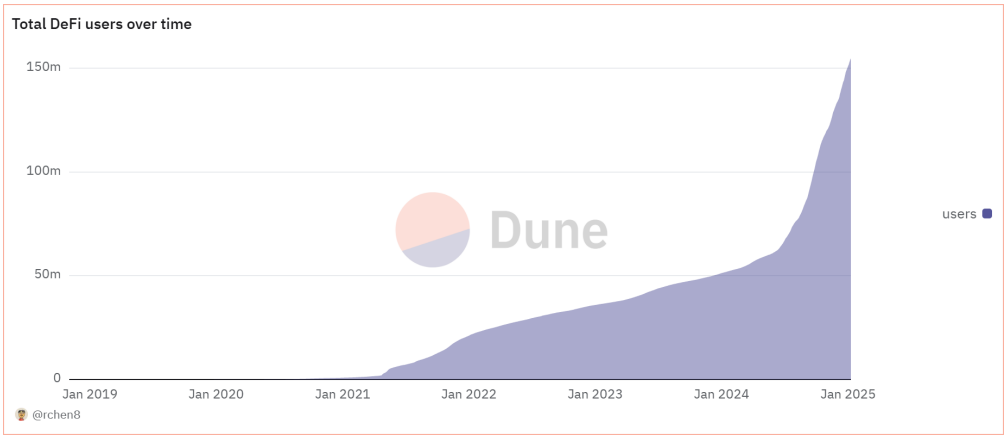


Figura 2.10. Utenti DeFi nel tempo

a disposizione degli utenti sono simili a quelli offerti dai sistemi finanziari tradizionali. I principali sono:

- **Monete stabili e pagamenti**: le stablecoin offrono un'alternativa meno volatile favorendo pagamenti digitali.
- **Scambio di asset digitali**: il trading decentralizzato permette lo scambio di criptovalute in modalità peer-to-peer senza passare per piattaforme centralizzate.
- **Prestiti e finanziamenti**: gli utenti possono prestare e prendere in prestito asset digitali senza bisogno di intermediari.
- **Rendimenti finanziari**: strategie come il farming e lo staking consentono agli utenti di ottenere guadagni attraverso la fornitura di liquidità o il blocco di fondi.
- **Copertura dei rischi**: le assicurazioni decentralizzate proteggono gli utenti da perdite dovute a vulnerabilità tecniche o problemi dei protocolli.
- **Derivati e strumenti avanzati**: la creazione e lo scambio di strumenti finanziari derivati consentono di replicare asset tradizionali in modo decentralizzato.

Tra le monete stabili sopracitate la più famosa è l'**USDC**, una criptovaluta collegata al dollaro USA tramite smart contract, che viene scambiata con un rapporto di 1 a 1. L'USDC appartiene alla categoria più recente di monete digitali chiamate **stablecoin**.
Le dApp maggiormente diffuse che si occupano di DeFi sono:

- **Uniswap**: piattaforma decentralizzata che permette agli utenti di scambiare diverse criptovalute utilizzando smart contract, senza la necessità di un'autorità centrale per determinare i tassi di cambio.
- **Compound**: piattaforma che sfrutta gli smart contract per permettere agli investitori di guadagnare interessi e ai richiedenti di ottenere prestiti istantanei, tutto senza l'intervento di un istituto bancario.

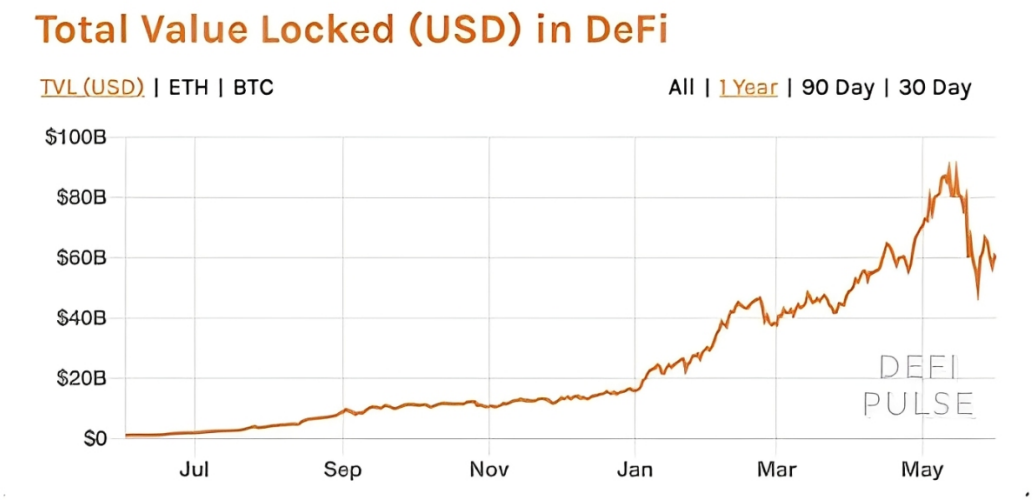


Figura 2.11. Capitalizzazione in dollari della DeFi — Maggio 2020-Giugno 2021

Queste applicazioni, che forniscono le condizioni ideali per lo sviluppo di attività finanziarie decentralizzate, sono dette **Decentralized Exchanges** (DEX). Questi ultimi sono stati introdotti con il fine di sostituire gli exchange centralizzati (CEX) che per molti anni hanno dominato lo scambio di criptovalute. Molti di essi, però, hanno subito disastrosi fallimenti e perdite, come è accaduto a novembre 2022 nel caso di FTX. Quest'ultimo era un importante exchange centralizzato, secondo solo a Binance, che ha dichiarato bancarotta a seguito di un buco finanziario di miliardi di dollari. Questo evento ha avuto ripercussioni devastanti sull'intero settore delle criptovalute, rafforzando la determinazione a far funzionare i DEX. Questo anche perché come evidenziato in diversi testi, ad esempio in Schär [2021] e Pourpouneh et al. [2020], gli exchange centralizzati presentano diversi svantaggi. I trader, infatti, devono avere piena fiducia nell'exchange poiché perdono la custodia degli asset, che rientrano nello stato patrimoniale dell'azienda, risultando aggredibili per risanare i debiti di un eventuale fallimento. i CEX sono inoltre suscettibili a minacce alla sicurezza a causa di un unico punto di attacco e alla mancanza di regolamentazione dettagliata.
I DEX, nonostante siano più complessi da implementare e possono portare a costi maggiori, rappresentano un sistema più sicuro per introdurre la finanza nel mondo blockchain. In aggiunta agli strumenti tradizionali, esistono strumenti più specifici per la DeFi, implementati anche dai DEX, come i **liquidity pool** che costituiscono l'elemento centrale dell'analisi trattata in questo lavoro. Essi vengono introdotti dettagliatamente nel capitolo 3.

Capitolo 3

Liquidity Pool

La DeFi, come introdotto nella sezione 2.6, ha avuto sempre più successo negli ultimi anni, rendendo necessaria l'introduzione di strumenti sempre più innovativi.
I **liquidity pool**, infatti, sono nati per rispondere alle esigenze di un mercato in crescita. Questi strumenti permettono agli utenti di depositare i propri fondi, offrendo loro l'opportunità di ottenere un guadagno potenziale. Il risultato è un insieme di criptovalute o token bloccati in uno smart contract, che vengono utilizzati per facilitare gli scambi tra gli asset su un DEX.
Prima di approfondire il loro funzionamento può essere utile evidenziare le analogie con il mercato tradizionale e identificare gli eventuali problemi da risolvere per ottenere una implementazione sicura ed efficiente.
Nel contesto della finanza tradizionale per riuscire ad abbinare domanda e offerta ci si serve degli **order book**, letteralmente libri degli ordini, facilitando così lo scambio degli asset.
Per poter introdurre lo stesso meccanismo nel mondo della DeFi, però, è necessario adattare l'uso degli order book in modo tale che possano funzionare anche in un ambiente senza intermediari centrali. Il ruolo che svolgono è lo stesso di quello dei libri degli ordini tradizionali, ma operando su una blockchain, sono in grado di garantire trasparenza e sicurezza.
Con l'adozione crescente della DeFi, gli order book decentralizzati stanno diventando uno strumento fondamentale per facilitare lo scambio di asset.
Esiste però un ostacolo da affrontare per introdurre gli order book in questo contesto. Essi, infatti, per poter funzionare adeguatamente devono poter registrare tutte le proposte di acquisto e vendita degli utenti su una blockchain pubblica, permettendo a chiunque di visualizzare gli ordini e garantendo un ambiente di trading più equo. Il fine è quello di utilizzare gli smart contract per abbinare gli ordini di acquisto e vendita automaticamente, senza la necessità di un'intermediazione di terze parti. Perché questo processo avvenga in modo corretto, gli order book decentralizzati devono conoscere i prezzi correnti degli asset.
Questo tipo di informazione, tuttavia, essendo soggetta a variazioni nel tempo, non può essere registrata e ottenuta in modo puntuale utilizzando solamente i dati all'interno della blockchain, introducendo così una nuova necessità: gli **oracoli**.

Questi ultimi sono servizi di terze parti che permettono di creare un ponte di informazioni tra la blockchain e il mondo esterno, dando la possibilità agli smart contract di accedere a informazioni aggiornate in tempo reale. Tra questi dati si possono trovare anche i prezzi correnti degli asset, rilevanti per il corretto funzionamento degli order book decentralizzati.
L'oracolo non è la vera e propria fonte del dato, ma rappresenta un servizio che raccoglie, verifica e autentica le informazioni provenienti da diverse fonti. L'esempio più famoso di questo tipo servizi è Chainlink [Breidenbach et al., 2021].
In questo modo, gli oracoli garantiscono che il processo di abbinamento degli ordini negli order book decentralizzati avvenga in modo efficiente, sicuro e affidabile, mantenendo l'integrità del mercato.
Il problema che si riscontra utilizzando questo metodo è associato al costo e all'affidabilità degli oracoli, in quanto le risorse richieste, per ottenere e verificare i dati esterni, sono notevoli. La veridicità dei dati, inoltre, è fondamentale, perché non venga compromesso il funzionamento degli smart contract e degli order book decentralizzati. Queste criticità possono derivare da vari fattori, come la malafede del servizio stesso, una sua vulnerabilità o la manipolazione delle informazioni da parte del cosiddetto "man-in-the-middle".
Per ovviare a questo problema, si possono adottare diverse strategie. Una delle soluzioni più efficaci è l'utilizzo degli AMM, ossia protocolli che automatizzano la fornitura di liquidità e la determinazione dei prezzi mediante formule matematiche, eliminando anche in questo caso la necessità di intermediari.
A differenza degli order book introdotti in precedenza, infatti, gli AMM utilizzano algoritmi per determinare i prezzi degli asset in modo automatico e continuo, così da non doversi affidare ad informazioni esterne. I Market Maker Automatizzati rappresentano dunque una delle innovazioni più importanti nel mondo della DeFi.
L'introduzione di questi strumenti innovativi è fondamentale al fine di comprendere il funzionamento dei liquidity pool. Gli utenti, quindi, possono depositare i loro token all'interno di questo sistema, che verranno gestiti automaticamente dall'AMM. I fornitori di liquidità vengono detti **LP**, e ciò che verseranno all'interno del pool rappresenterà la riserva totale di valore del protocollo. Solitamente, gli LP per incrementare la liquidità del pool sono obbligati a versare una combinazione specifica di token in una proporzione fissa. Questo concetto verrà approfondito meglio successivamente in questo capitolo. Uno degli incentivi che si ha nel fornire criptovalute a questi sistemi è il diritto a una quota delle commissioni pagate dai trader per lo scambio dei token in proporzione al totale della liquidità presente nel pool.
Come introdotto in precedenza, gli AMM sono in grado di determinare i prezzi di mercato in modo algoritmico. Ciò è possibile grazie alle opportunità di arbitraggio che vengono a crearsi in questo contesto. A livello matematico l'arbitraggio si definisce nel seguente modo

Definizione: 3.0.1. *Un'opportunità di arbitraggio in un mercato finanziario è un portafoglio autofinanziato p tale che:*

$$V^p(0) = 0$$
$$\mathbb{P}(V^p(T) \geq 0) = 1$$
$$\mathbb{P}(V^p(T) > 0) > 0$$

Un modello di mercato finanziario tradizionale di solito si considera completo se non ammette la possibilità di arbitraggi al suo interno [Björk, 2009]. Nella DeFi, invece, è ciò che permette il funzionamento accurato degli AMM. Si sfrutterà l'introduzione di uno specifico AMM per poter approfondire questo meccanismo — Sezione 3.1.1.
Esistono diversi tipi di AMM che sfruttano algoritmi differenti per calcolare correttamente il prezzo del proprio mercato. Verranno introdotti dunque:

- AMM a prodotto costante;
- AMM a media costante;
- AMM a somma costante;
- AMM ibridi;
- AMM a liquidità concentrata;
- Bootstrap AMM.

Questi sono i principali AMM implementati attualmente sulle piattaforme di DeFi. Grazie al fatto che seguono formule matematiche ben precise, è stato possibile teorizzarne molteplici, che però non sono mai stati realmente introdotti, in quanto non coerenti con l'andamento reale del mercato.

3.1 Tipologie di AMM

Nella sezione seguente vengono introdotti i principali tipi di AMM che saranno utili nel corso dei capitoli successivi. A tal fine vengono utilizzate definizioni ed esempi tratti da Mohan [2022].

3.1.1 AMM a prodotto costante

Il modello a **prodotto costante** (CPMM), viene implementato in liquidity pool in cui è presente una coppia di token e la proporzione tra i due è del tipo 50/50, ossia il valore fornito deve essere suddiviso equamente tra i due token. L'AMM ha, dunque, l'obiettivo di mantenere costante il prodotto tra due token x e y, nel seguente modo:

$$x \cdot y = k \tag{3.1}$$

Nella formula 3.1, x indica la quantità del primo token, y la quantità del secondo e k rappresenta l'invariante, ossia il valore che l'AMM vuole mantenere costante. L'andamento è rappresentato nel grafico in figura 3.1.
Per comprendere a pieno questo meccanismo risulta utile introdurre un esempio pratico, tratto da Mohan [2022].
I due token in esame sono il token X, che si assume essere ETH, e il token Y, un qualsiasi token ERC-20. Di seguito le lettere maiuscole saranno utilizzate per indicare il token in generale e lettere minuscole per indicarne la quantità specifica.

Per qualsiasi quantità di token x, y, un CPMM utilizza la funzione di scambio 3.1 per gestire algoritmicamente il commercio tra i due token.
Si supponga che al momento dell'investimento il valore dell'ETH sia pari a 50 USD e quello del token y pari a 1 USD, come per le stablecoin. La quantità iniziale di ETH fornita dagli LP è $x_0 = 10$, e la quantità di stablecoin fornita è $y_0 = 500$. Pertanto, il valore cumulato dei token sarà pari a 500 USD e rappresenterà la riserva iniziale nella liquidity pool dell'AMM.
In un CPMM, il prodotto dei due è una costante o invariante, k, che assume un valore iniziale:

$$k_0 = x_0 \cdot y_0 = 5000.$$

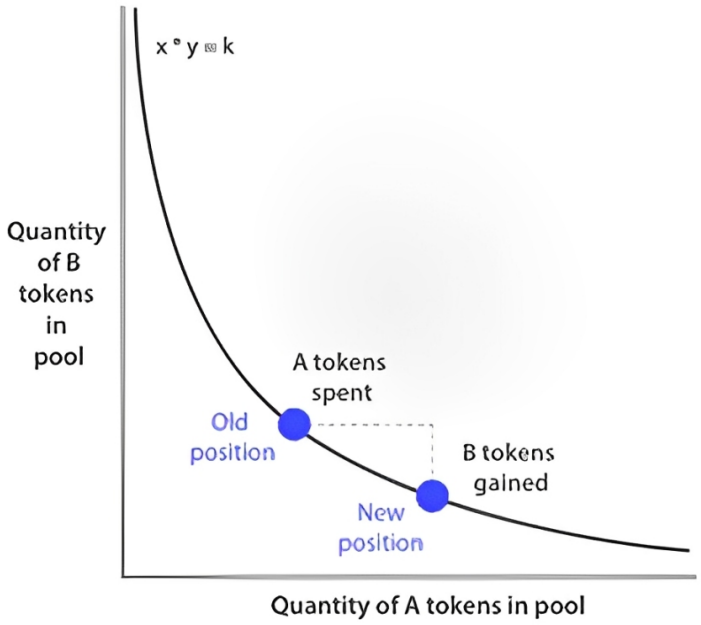


Figura 3.1. Grafico rappresentante la formula dei CPMM[1]

In questo contesto è fondamentale il ruolo dei **trader**, ossia un qualsiasi agente che scambia un token con un altro nella pool. A tale scopo il trader deve pagare una commissione di trading, che viene accreditata agli LP. Tuttavia, per semplicità, consideriamo prima la situazione in cui la commissione di trading è zero. Supponiamo che si desideri vendere 1 ETH in cambio di stablecoin. In assenza di una commissione di trading, l'intera quantità viene aggiunta alla liquidity pool, risultando in un saldo della pool di $x_1 = 11$ ETH. Dato

[1]Fonte: https://whiteboardcrypto.com/impermanent-loss-calculator/.

k_0, la nuova quantità di token Y, y_1, nel pool sarà:

$$y_1 = \frac{k_0}{x_1} = \frac{5{,}000}{11} = 454.55.$$

La variazione del secondo token nelle riserve del pool è $y_1 - y_0 = 454.55 - 500 = -45.45$, è negativa perché le riserve nell'AMM sono diminuite; questa sarà la quantità di token inviata all'account del trader. Il trader ha effettivamente venduto 1 ETH per acquistare 45.45 token Y con l'AMM, di conseguenza, i fornitori di liquidità ora detengono in aggregato 1 ETH in più e 45.45 stablecoin in meno. Sebbene la riserva sia stata alterata, è evidente che $x_1 \cdot y_1 = k_0$, e quindi il prodotto delle nuove quantità è uguale all'invariante. È altrettanto semplice lavorare sull'esempio in senso inverso, dove il trader acquista 1 ETH dall'AMM. Procedendo come prima: $x_1 = 9$, dunque

$$y_1 = \frac{k_0}{x_1} = \frac{5{,}000}{9} = 555.56$$

e di conseguenza $y_1 - y_0 = 555.56 - 500 = 55.56$.
Il prezzo nel primo caso sarà $P^a_{\frac{Y}{X}} = 55.56$, e rappresenterà il prezzo **ask** di ETH ossia il prezzo al quale l'AMM **vende** 1 unità di X al trader. Nel secondo invece quello che si ottiene è il prezzo **bid**, ovvero il prezzo al quale l'AMM **acquista** ETH. Dai calcoli si osserva che questa quantità è pari a $P^b_{\frac{Y}{X}} = 45.45$
I tassi bid e ask in questo esempio sembrano sensibilmente diversi, infatti, lo spread bid-ask è $P^a_{\frac{Y}{X}} - P^b_{\frac{Y}{X}} = 55.56 - 45.45 = 10.10$, principalmente perché i cambiamenti considerati sono significativi e di grande entità in proporzione alla liquidità presente nel pool al momento dello scambio, per semplificare l'esempio. In contesti reali, ETH è divisibile fino a 18 cifre decimali, quindi sono possibili operazioni molto più piccole.
Per quanto riguarda variazioni infinitesimali si può dimostrare che i due prezzi convergono:

$$P^a_{\frac{Y}{X}} = P^b_{\frac{Y}{X}} = \frac{y_0}{x_0} = P_{\frac{Y}{X}} \tag{3.2}$$

Se venissero introdotte le fee nell'esempio si osserverebbe che la differenza tra il prezzo bid e il prezzo ask aumenterebbe ulteriormente, rendendo necessaria una modifica all'equazione 3.2.
Osservando il grafico in figura 3.1 si nota che, dunque, per cambiamenti infinitesimali si avrà:

$$P_{\frac{Y}{X}} = \frac{y}{x} = \frac{dy}{dx} \tag{3.3}$$

L'equazione 3.3 rappresenta la pendenza della retta tangente al grafico, che varia però, in ogni punto, rendendolo poco stabile. Quando uno dei due token tende a zero, quindi, si avrà che il prezzo tenderà a infinito. Ciò invece non avviene al variare di k, infatti, poiché la formula 3.1 è omogenea.

Definizione: 3.1.1. *Una funzione $f(x_1, ..., x_n) \in \mathbb{R}^n$ si dice omogenea di grado k se $\forall \alpha > 0$ e per ogni scelta di variabili $x_1, ..., x_n$: $f(\alpha x_1, ..., \alpha x_n) = \alpha^k f(x_1, ..., x_n)$.*

Più semplicemente, tutti i punti appartenenti allo stesso raggio dall'origine, corrisponderanno allo stesso prezzo.
Per comprendere le conseguenze di queste osservazioni è necessario introdurre il concetto di **slippage**:

Definizione: 3.1.2. *lo slippage è la differenza tra il prezzo di esecuzione di un ordine rispetto a quello richiesto dall'operatore, si verifica quando un mercato si muove troppo velocemente.*

Si osserva che questo fenomeno si verifica quando avvengono grandi transazioni eseguite dai trader, ma non quando viene inserita liquidità all'interno del pool.
La fornitura di liquidità condotta in questo modo cambia la curva di livello da k_0 a k_1 senza far variare il prezzo.
Come evidenziato in precedenza l'arbitraggio è fondamentale per il funzionamento degli AMM, ciò perché i prezzi del mercato interno siano sempre allineati con quelli esterni.
Supponendo che il prezzo esterno di mercato sia $M_{\frac{Y}{X}} > P_{\frac{Y}{X}}$, si crea un'opportunità di arbitraggio per i trader: essi possono, infatti, comprare il token X ad un prezzo inferiore all'interno del pool, per poi rivenderlo ad un prezzo maggiore esternamente. In questo modo, tale token verrà acquistato dagli arbitraggisti, finché la diminuzione dello stesso porterà il prezzo $P_{\frac{Y}{X}}$ a tendere a quello esterno $M_{\frac{Y}{X}}$. Analogamente accade nella situazione opposta, se $M_{\frac{Y}{X}} < P_{\frac{Y}{X}}$ i trader acquisteranno il token sul mercato esterno per ottenere un profitto rivendendolo all'interno del liquidity pool. In questo caso la quantità aumenterà facendo sì che il prezzo diminuisca fino alla condizione di non arbitraggio $M_{\frac{Y}{X}} = P_{\frac{Y}{X}}$.
Un semplice esempio numerico permette di comprendere meglio questo meccanismo[2]. Si supponga di prendere in considerazione un liquidity pool con le seguenti caratteristiche:

Token	**Valore in USD**	**Quantità**	**Totale in USD**
DAI	\$1	10,000	\$10,000
ETH	\$500	20	\$10,000

Tabella 3.1. Quantità token nel liquidity pool

Chiaramente in questo caso l'invariante k sarà pari a

$$x \cdot y = 20 \cdot 10{,}000 = 200{,}000 \ ,$$

dove x rappresenta la quantità di ETH e y la quantità di DAI, una stablecoin.
Se il prezzo dell'ETH salisse a \$550 nel mercato esterno si creerebbe un'opportunità di arbitraggio riconducibile alla prima casistica trattata. I trader, infatti, per sfruttare questa

[2]L'esempio è tratto dal canale YouTube di Finematics. Fonte: https://www.youtube.com/watch?v=cizLhxSKrAc.

condizione dovrebbero acquistare ETH all'interno del pool per poi rivenderli, guadagnandoci, sul mercato esterno.
Si osserva che, se inizialmente $P_{\frac{Y}{X}}$ è uguale a \$500, a seguito del cambio di prezzo dell'ETH sarà pari a \$550. Da questa relazione si può ricavare che:

$$y = 550x.$$

Sostituendo nell'equazione del CPMM si ottiene:

$$550x^2 = 200{,}000\ ,$$

da cui si può ricavare x, che rappresenta la nuova quantità di ETH. Le nuove proporzioni saranno, dunque:

$$x = \sqrt{\frac{200{,}000}{550}} = 19.07$$
$$y = \frac{200{,}000}{19.07} = 10{,}487.68.$$

Gli arbitraggisti hanno acquistato 0.93 ETH all'interno del pool, per poterli rivendere esternamente ad un prezzo maggiore, facendo sì che i due prezzi si allineassero nuovamente.
Il DEX di maggiore successo che ha implementato questo tipo di AMM è Uniswap. La prima versione è stata lanciata nel 2018, ed è nota come v1. Questa versione è stata successivamente sostituita nel 2020 da Uniswap v2. La principale differenza è che nella versione più recente le liquidity pool possono contenere qualsiasi coppia di token, a patto che rispettino lo standard di Ethereum ERC-20. Nella prima versione, invece, l'ETH doveva necessariamente essere incluso nella coppia.
Questo ha permesso di facilitare gli scambi tra token differenti, eliminando il passaggio strettamente necessario attraverso l'ETH. Ciò ha anche aperto le porte ad un nuovo tipo di arbitraggio più complesso, detto arbitraggio triangolare: i trader posso guadagnare partendo da un token X scambiandolo per Y, successivamente per Z, per poi tornare a X.

3.1.2 AMM a somma costante

Gli AMM a **somma costante** (CSMM) fanno parte degli AMM introdotti a livello puramente teorico, in quanto attualmente, a causa delle limitazioni che verranno descritte, non sono stati implementati da nessun DEX.
L'algoritmo di seguito prevede che in questo caso la quantità mantenuta costante sia la somma delle riserve. La funzione è la seguente:

$$x + y = k \tag{3.4}$$

Nella formula 3.4, x indica nuovamente la quantità del primo token, y la quantità del secondo e k rappresenta l'invariante, ossia il valore che l'AMM vuole mantenere costante, come visto per il caso CPMM — equazione 3.1. Questa formula può essere generalizzata

per essere utilizzata in pool di liquidità in cui è possibile inserire più di due token, anche in proporzioni differenti, affinché:

$$\sum_{i=1}^{n} a_i x_i = k \tag{3.5}$$

Dove x_i rappresenta la quantità dell'i-esimo token e a_i il peso ad esso assegnato. Il problema di questo tipo di AMM è il fatto che non risulta ottimale come exchange, perché i trader, svolgendo attività di arbitraggio, potrebbero svuotare completamente il liquidity pool, lasciando solamente uno dei due token. Prendendo in considerazione il caso con due token è evidente questa problematica. È sufficiente, infatti, concentrarsi sul grafico in figura 3.2 per osservare come i token possano liberamente raggiungere lo zero, cosa che non accade nel caso a prodotto costante in cui i token possono tendere a zero, ma mai annullarsi completamente.

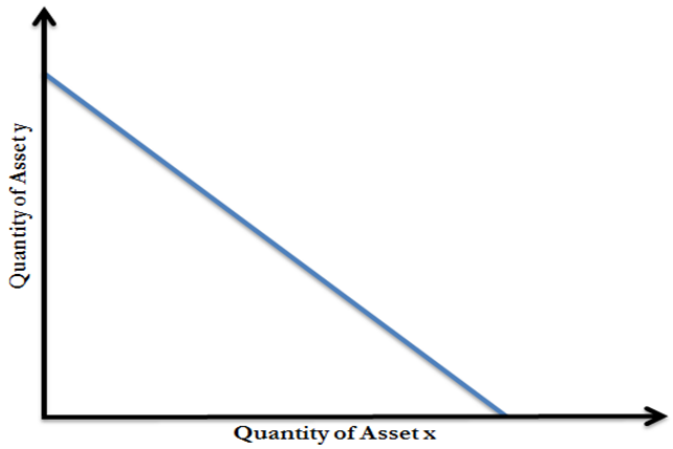


Figura 3.2. Grafico rappresentante la formula dei CSMM[3]

Per quanto questo risulti essere un problema nell'implementazione concreta, è comunque interessante dal punto di vista teorico. Ciò perché può portare allo sviluppo di AMM, come ad esempio quelli ibridi, che sfruttino solo in parte questo algoritmo per trarne i lati positivi.
Lo svantaggio maggiore degli AMM a prodotto costante, presentati nella Sezione 3.1.1, è il fatto che il prezzo vari ad ogni transazione. Questo, però, non avviene per quanto riguarda l'algoritmo della somma costante. Considerando una semplificazione della formula generale $ax + by = k$, si osserva che il prezzo è semplicemente pari alla pendenza della retta, che non cambia al variare delle riserve. Formalmente, il prezzo $P_{\frac{Y}{X}}$ è pari a $-\frac{dy}{dx} = \frac{a}{b}$, ossia una quantità costante. Nel caso in cui i due pesi fossero uguali il rapporto che si otterrebbe sarebbe pari a 1, sottolineando che lo scambio dei due token avviene alla pari. Questa caratteristica è molto importante perché non crea problemi per lo scambio

[3]Fonte: https://hackernoon.com/automated-market-makers-what-you-need-to-know.

di token con prezzo stabile, ovvero le stablecoin.
Si trasforma però in un problema quando il prezzo è variabile sul mercato esterno. Nel caso più semplice, come spiegato in precedenza, si ha $P_{\frac{Y}{X}} = 1$. Se esternamente al pool $M_{\frac{Y}{x}} > 1$, gli arbitraggisti sfrutteranno questa disparità per ottenere un profitto pari a $M_{\frac{Y}{X}} - P_{\frac{Y}{X}}$ per ogni unità di token acquistata all'interno del pool e rivenduta su un mercato esterno. Lo stesso meccanismo si attiverebbe nel caso in cui $M_{\frac{Y}{x}} < 1$. La variazione della quantità di token in questo tipo di AMM però non porta ad un conseguente cambio di prezzo, rendendo sfruttabile l'opportunità di arbitraggio fino all'esaurimento di uno dei due token.
Si può concludere che una soluzione di tale genere è meno vantaggiosa per token con prezzi volatili, perché come detto in precedenza, per sfruttare le opportunità di arbitraggio, i trader potrebbero svuotare il pool.

3.1.3 AMM a media costante

Gli AMM a **media costante** (CMMM) sono una generalizzazione del caso a prodotto costante — Sezione 3.1.1. In questo caso, infatti, l'algoritmo prevede che sia la seguente quantità a rimanere costante:

$$\begin{aligned} &\prod_{i=1}^{n} x_i^{w_i} = k \\ con \quad &\sum_{i=1}^{n} w_i = 1 \end{aligned} \tag{3.6}$$

Dove, x_i rappresa le quantità dei diversi token, inserite all'interno del pool con una data proporzione w_i. Si osserva facilmente come il caso del prodotto costante è analogo ad avere $i = 2$ e $w_1 = w2 = \frac{1}{2}$.
Anche in questo caso, per semplicità, si prende in esempio il caso senza commissioni. Per ricavare i prezzi relativi dei token si procede in modo analogo a quelli precedenti per cui $P_{\frac{x_i}{x_j}} = \frac{-dx_i}{dx_j}$, che partendo dall'equazione 3.6, risulta essere:

$$P_{\frac{x_i}{x_j}} = -\frac{dx_i}{dx_j} = \frac{w_j}{w_i}\frac{x_i}{x_j} \tag{3.7}$$

Questo tipo di algoritmo è implementato per gestire gli AMM sui DEX di Balancer [Martinelli and Mushegian, 2019]. Questi DEX offrono, infatti, la possibilità di investire in pool multi token o in pool non bilanciate.

3.1.4 AMM ibridi

Gli AMM **ibridi** (HFMM) nascono con l'intento di combinare gli aspetti positivi degli AMM a prodotto costante e quelli a somma costante. L'algoritmo che seguono questo tipo di AMM deriva direttamente dalle formule 3.4 e 3.6, in cui in entrambi i casi i pesi

vengono considerati tutti uguali tra loro, pari a $\frac{1}{n}$. Pertanto, la formula che si ottiene è:

$$\lambda \frac{\sum_{i=1}^{n} x_i}{n} + (1 - \lambda) \prod_{i=1}^{n} x_i^{\frac{1}{n}} = k \tag{3.8}$$

Il parametro λ è un valore compreso tra 0 e 1. L'obiettivo è quello di dare un peso maggiore alla somma quando le due quantità risultano bilanciate, viceversa un peso maggiore al prodotto costante quando uno dei due token rischia di raggiungere lo zero. Per poterne comprendere il significato grafico è stata inserita la figura 3.3, in cui si evidenziano le differenze con un AMM a prodotto costante e uno a somma costante.

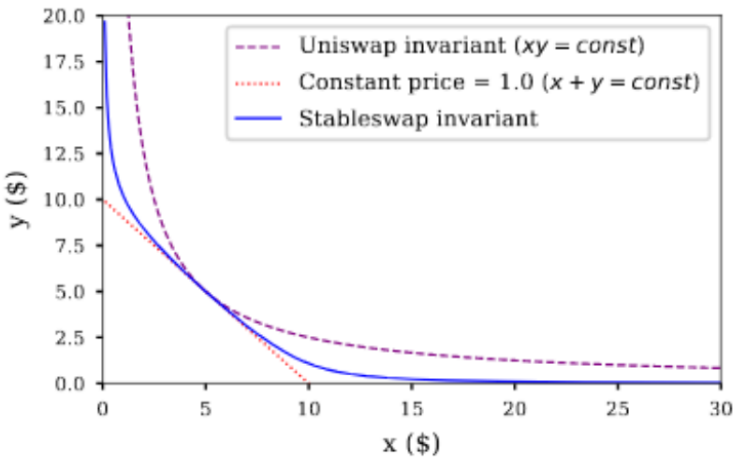


Figura 3.3. Grafico rappresentante la formula dei HFMM[4]

Anche in questo caso può essere scritto il prezzo relativo $P_{\frac{x_i}{x_j}}$ esplicitamente, per poter successivamente comprendere il legame tra λ e la composizione del pool. A tal fine vengono introdotte le seguenti notazioni:

$$A = \frac{\sum_{i=1}^{n} x_i}{n} \tag{3.9}$$

$$G = \left(\prod_{i=1}^{n} x_i \right)^{\frac{1}{n}} \tag{3.10}$$

Dove l'equazione 3.9 rappresenta la media aritmetica (AM) delle riserve e la 3.10 la media geometrica (GM). Ricordando la relazione che intercorre tra AM e GM:

[4]Fonte: https://medium.com/@DailyUniverse_/uniswap-v3%E7%9A%84%E7%82%BC%E9%87%91%E7%A7%98%E6%9C%AF-33169342b923.

Teorema: 3.1.1. *Per ogni $x_1, ..., x_n \in \mathbb{R}$ non negativi vale la seguente disuguaglianza:*

$$\frac{\sum_{i=1}^{n} x_i}{n} \geq \left(\prod_{i=1}^{n} x_i \right)^{\frac{1}{n}}$$

dove l'uguaglianza si verifica se e solo se $x_1 = x_2 = \cdots = x_n$

Sfruttando il teorema 3.1.1, come svolto in [Mohan, 2022], si osserva che concentrandosi su $\frac{G}{A}$, con $A \neq 0$, si avrà che $0 \leq \frac{G}{A} \leq 1$. Ciò rende questa quantità un candidato ideale per rappresentare il parametro λ, poiché fornisce una misura coerente della transizione tra i diversi modelli di market making.
Sostituendo questa quantità in 3.8 si ottiene:

$$G \cdot \left(2 - \frac{G}{A} \right) = k \tag{3.11}$$

Di conseguenza $P_{\frac{x_i}{x_j}}$ sarà:

$$P_{x_i/x_j} = -\frac{dx_i}{dx_j} = \frac{2(1 - \frac{G}{A})\frac{\partial G}{\partial x_j} + \frac{G^2}{A^2}\frac{\partial A}{\partial x_j}}{2(1 - \frac{G}{A})\frac{\partial G}{\partial x_i} + \frac{G^2}{A^2}\frac{\partial A}{\partial x_i}} \tag{3.12}$$

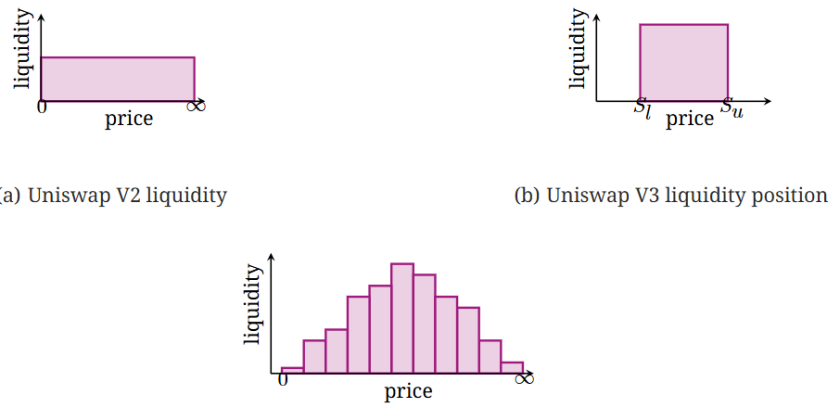
Dalla formula 3.12 è chiaro che se uno dei token tendesse a zero allora $\frac{G}{A} \to 0$ e $P_{\frac{x_i}{x_j}} \to \frac{\frac{\partial G}{\partial x_j}}{\frac{\partial G}{\partial x_i}}$, che è analoga a 3.7, sottolineando come prevalga il CMMM. Il risultato è analogo se $\frac{G}{A} \to 1$, dunque $P_{\frac{x_i}{x_j}} \to \frac{\frac{\partial A}{\partial x_j}}{\frac{\partial A}{\partial x_i}}$ che rappresenta, invece, il prezzo nel caso CSMM.
Questo è solo uno dei possibili modi di sfruttare l'AMM ibrido. Può essere definita, infatti, una funzione più generale detta **Constant Elasticity of Substitution** (CES):

$$E\left[\alpha x^{\rho} + (1 - \alpha) y^{\rho}\right]^{\frac{1}{\rho}} \tag{3.13}$$

Dove E è un parametro di efficienza, α è un parametro di distribuzione e ρ è un parametro di sostituzione.
Questo approccio offre un ulteriore grado di flessibilità, consentendo di modificare la curvatura della funzione di scambio in modo parametrico, similmente alla regolazione di un peso fisso λ.

3.1.5 AMM a liquiidtà concentrata

Gli **AMM a liquidità concentrata** sono un'ulteriore variante dei CPMM. In questo caso, infatti, l'algoritmo di base funziona mantenendo il prodotto costante, ma gli LP hanno la possibilità di scegliere l'intervallo su cui investire, dato da $[P_{min}, P_{max}]$. Nei CPMM, la liquidità fornita dagli utenti viene distribuita uniformemente lungo l'intera curva dei prezzi, anche in fasce in cui gli scambi sono rari o inesistenti, causando una dispersione inefficiente del capitale.
In questo caso invece è possibile concentrare la liquidità nei range in cui si verifica la



(a) Uniswap V2 liquidity

(b) Uniswap V3 liquidity position

(c) collection of Uniswap V3 liquidity positions

Figura 3.4. Rappresentazione dell'efficienza del capitale [Heimbach et al., 2022]

maggior parte delle transazioni, aumentando l'efficienza del capitale, come si può osservare dall'immagine 3.4.
Di conseguenza gli LP possono ottenere rendimenti più elevati con un capitale inferiore, poiché una maggiore frazione della loro liquidità viene effettivamente utilizzata per gli scambi. Si è però esposti ad un rischio maggiore, poiché quando $P_{\frac{Y}{X}} \notin [P_{min}, P_{max}]$, i fornitori di liquidità non ricevono più le commissioni di trading, dato che la loro liquidità non è più attivamente utilizzata negli scambi. Ciò è evidente anche a livello grafico, della figura 3.5, in cui la curva di attività risulta essere una traslazione di quella rappresentata per il CPMM — figura 3.1.
Per ridurre questo rischio gli LP devono gestire attivamente la loro liquidità, in modo da riallocarla strategicamente in base ai cambiamenti di prezzo, creando posizioni uniche.
Questa osservazione può essere tradotta matematicamente per comprendere la quantità effettiva di token posseduti degli LP nel seguente modo:

$$\tilde{x}_{\text{real}} = \begin{cases} \tilde{L} \cdot \left(\frac{1}{P_{min}} - \frac{1}{P_{max}} \right) & \text{se } P_{\frac{Y}{X}} < P_{min} \\ \tilde{L} \cdot \left(\frac{1}{P_{\frac{Y}{X}}} - \frac{1}{P_{max}} \right) & \text{se } P_{min} \leq P_{\frac{Y}{X}} < P_{max} \\ 0 & \text{se } P_{\frac{Y}{X}} \geq P_{max} \end{cases}$$

$$\tilde{y}_{\text{real}} = \begin{cases} 0 & \text{se } P_{\frac{Y}{X}} < P_{min} \\ \tilde{L} \cdot (P_{\frac{Y}{X}} - P_{min}) & \text{se } P_{min} \leq P_{\frac{Y}{X}} < P_{max} \\ \tilde{L} \cdot (P_{max} - P_{min}) & \text{se } P_{\frac{Y}{X}} \geq P_{max} \end{cases} \tag{3.14}$$

Dove $\tilde{L}$ rappresenta la liquidità inserita dagli LP nel pool, nell'intervallo stabilito. Quando il prezzo è all'interno dell'intervallo di prezzo scelto, il fornitore di liquidità detiene entrambi i token nel pool e la sua liquidità è attiva, al contrario quando il prezzo è inferiore o superiore ai limiti, avrà solo uno dei due token.
Si osserva che intervalli più ampi riducono l'efficienza del capitale, ma riducono il rischio di renderlo inattivo.
Nei DEX che decidono di implementare questo tipo di AMM, possono dunque esistere più pool contenenti la stessa coppia di token, con livelli di fee differenti, per permettere di far fronte ai rischi diversi che si hanno su intervalli più o meno ampi. Le commissioni inoltre non vengono reinvestite, ma consegnate direttamente agli LP.
L'DEX di maggior rilievo in cui è stato implementato questo tipo di AMM è Uniswap v3, lanciata nel 2021 [Adams et al., 2021].

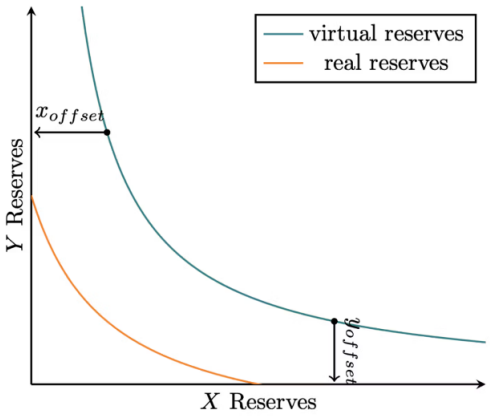


Figura 3.5. Differenza tra prodotto costante e liquidità concentrata [5]

3.1.6 Bootstrap AMM

I **bootstrap** AMM (BAMM) sono una variante degli AMM tradizionali progettata per inizializzare il prezzo di un token nelle prime fasi di un protocollo DeFi, spesso utilizzati per nuovi token o pool emergenti. A differenza degli AMM standard, che richiedono una liquidità iniziale significativa per funzionare efficacemente, i BAMM adottano meccanismi che incentivano i fornitori di liquidità in modo progressivo, riducendo il rischio di manipolazioni da parte di grandi investitori, detti **whale**. Essi hanno il potere di influenzare il mercato con le loro operazioni, poiché possono muovere grandi volumi di liquidità in una singola transazione, causando forti oscillazioni di prezzo, problemi di slippage e, in alcuni casi, manipolazioni di mercato come il *pump and dump* (gonfiare e poi scaricare un asset).

[5] Fonte: https://www.paradigm.xyz/2021/06/uniswap-v3-the-universal-amm.

I BAMM vengono utilizzati per l'**auction-based pricing**, il cui scopo è quello di determinare il prezzo di un dato token dinamicamente in base alla domanda e all'offerta iniziale, evitando problemi di alta volatilità e manipolazioni di mercato. Si consente, così, un accesso equo alla liquidità senza forti oscillazioni iniziali.
La strategia che viene adottata è quella di impostare un prezzo iniziale elevato per il token, così che i bot o i grandi investitori siano disincentivati da acquistarne grandi quantità, riducendo il rischio di speculazioni rapide. Questo processo viene applicato a coppie costituite da un token di apprezzamento e una stablecoin, in modo da poter sfruttare la stabilità di quest'ultima.
Risulta fondamentale, inoltre, la scelta dei pesi iniziali della composizione del pool: solitamente si parte da un rapporto sbilanciato, ad esempio 95% rappresentato dal token e 5% da stablecoin. Questo approccio consente una maggiore stabilità nella fase iniziale, riducendo la volatilità del prezzo del token e facilitando una distribuzione più graduale ed equilibrata nel tempo.
Il bilanciamento delle quantità all'interno della pool avviene a intervalli di tempo regolari, variando progressivamente la distribuzione degli asset. Questo processo continua fino a raggiungere un equilibrio tra i due asset, garantendo una scoperta del prezzo più fluida e naturale. Questo ribilanciamento graficamente può essere rappresentato come in figura 3.6.

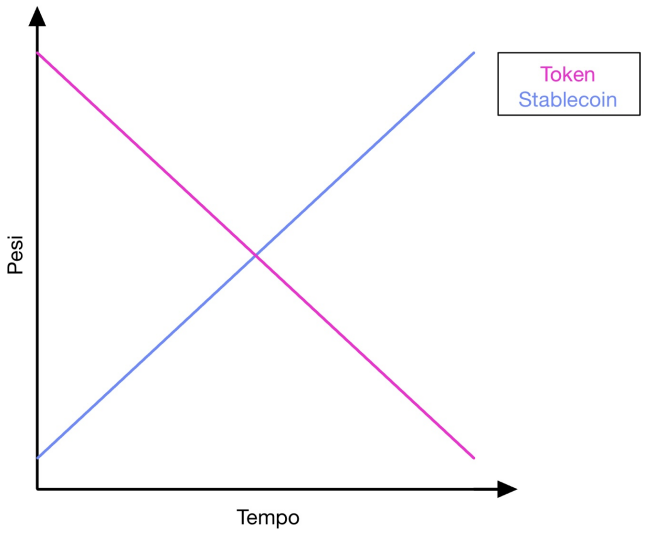


Figura 3.6. Ribilanciamento bootstrap AMM

Una volta stabilito il prezzo ottimale per il mercato, l'AMM inizia a funzionare come un CPMM.
I bootstrap AMM rappresentano, dunque, un'evoluzione importante nel settore della DeFi, offrendo meccanismi più sofisticati per il lancio di nuovi asset e migliorando la sostenibilità a lungo termine della liquidità nei mercati emergenti. Questo approccio, infatti, non solo

permette una scoperta del prezzo più equa e una maggiore stabilità nella fase di lancio di un nuovo mercato decentralizzato, ma aiuta anche a mitigare problemi di IL e slippage elevato tipici dei pool con bassa liquidità iniziale.

3.2 Impermanent loss

L'**IL** è uno dei principali problemi dei liquidity pool. I ribilanciamenti dovuti agli algoritmi che seguono gli AMM fanno sì che si possa verificare questo tipo di perdita.
Specificamente, l'IL è un fenomeno che si verifica negli AMM quando il prezzo relativo dei due asset in un pool cambia rispetto al momento in cui la liquidità è stata depositata. Viene utilizzata per rappresentare la differenza di valore tra mantenere gli asset in un pool di liquidità (V_{POOL}) e detenerli semplicemente in un wallet (V_{HOLD}).
Questa perdita si verifica implementando AMM a prodotto costante, a liquidità concentrata o ibridi, poiché il protocollo riequilibra automaticamente le quantità di asset nel pool, portando gli LP a detenere proporzioni diverse rispetto alla loro allocazione iniziale. Nei primi due casi può essere facilmente determinata da una formula, nell'ultimo caso sono invece necessari metodi numerici più complessi.
La formula che si ottiene nel caso di un CPMM è la seguente:

$$IL = 1 - \frac{V_{POOL}}{V_{HOLD}} = 1 - \frac{2 \cdot \sqrt{p}}{1+p}. \tag{3.15}$$

L'equazione 3.15 può essere dimostrata:

Dimostrazione 3.2.1. *Si supponga che un LP depositi due asset x e y nel pool con un rapporto iniziale dato da:*

$$\frac{y_0}{x_0} = p_0$$

dove p_0 è il prezzo iniziale dell'asset x rispetto all'asset y. Si supponga ora che il prezzo cambi da p_0 a un nuovo valore p_1, quindi:

$$\frac{y_1}{x_1} = p_1$$

Poiché la funzione del pool è $x \cdot y = k$, le nuove quantità nel pool diventano:

$$x_1 = \sqrt{\frac{k}{p_1}}, \quad y_1 = \sqrt{kp_1}$$

Il valore della liquidità nel pool dopo la variazione di prezzo è:

$$V_{POOL} = x_1 p_1 + y_1$$

Sostituendo x_1 e y_1:

$$V_{POOL} = \sqrt{\frac{k}{p_1}} \cdot p_1 + \sqrt{kp_1} = \sqrt{kp_1} + \sqrt{kp_1} = 2\sqrt{kp_1}$$

Se l'LP avesse semplicemente detenuto gli asset senza fornirli al pool, il valore sarebbe:

$$V_{HODL} = x_0 p_1 + y_0$$

Poiché $x_0 = \frac{y_0}{p_0}$*, può essere riscritto come:*

$$V_{HODL} = \left(\frac{y_0}{p_0}\right) p_1 + y_0 = y_0 \left(\frac{p_1}{p_0} + 1\right) = 2y_0 \frac{p_1}{p_0}$$

L'IL è definita come la perdita relativa:

$$IL = 1 - \frac{V_{POOL}}{V_{HODL}}$$

Sostituendo i valori calcolati:

$$IL = 1 - \frac{2\sqrt{kp_1}}{2y_0 \frac{p_1}{p_0}}$$

Poiché $k = x_0 y_0 = \frac{y_0^2}{p_0}$*, si può semplificare:*

$$IL = 1 - \frac{2\sqrt{y_0^2 \cdot p_1/p_0}}{2y_0 \cdot p_1/p_0} = 1 - \frac{2y_0\sqrt{p_1/p_0}}{2y_0 \cdot p_1/p_0} = 1 - \frac{2\sqrt{p}}{1+p}$$

Dove $p = \frac{p_1}{p_0}$*, ottenendo la formula finale:*

$$IL = 1 - \frac{2\sqrt{p}}{1+p}$$

□

È stato dimostrato che l'IL dipende direttamente dalla variazione del prezzo tra gli asset nel pool. Se il prezzo ritorna al valore iniziale, la perdita scompare.
Proprio per questo motivo tale perdita è detta *impermanente*: non viene effettivamente capitalizzata ad ogni cambiamento di prezzo, ma solo quando la liquidità viene ritirata dal pool. Se l'LP, tuttavia, ritira la liquidità prima che il prezzo si riequilibri, l'IL diventa una perdita effettiva.
Il valore dell'IL aumenta, dunque, con la **volatilità**: più il prezzo si discosta dal valore originale, maggiore sarà la perdita relativa. Segue un semplice esempio numerico derivato dall'esempio nella sezione 3.1.1. Prendendo sempre in considerazione i dati relativi alla tabella 3.1, si supponga che un LP depositi 2 ETH dal valore di $500 e 1000 DAI, contribuendo al pool con un valore pari a $2,000. Questa cifra rappresenta il 10% del valore totale ($20,000) all'interno di questa pool.
Si consideri nuovamente che il prezzo dell'ETH passi da $500 a $550, di conseguenza, dai dati dell'esempio precedente le nuove quantità saranno 19.07 ETH e 10487.68 DAI. Il fornitore di liquidità, di conseguenza, avrà 1.907 ETH e 1048.768 DAI, che corrispondono a un valore di:

$$1.907 \cdot 550 + 1{,}048.768 \cdot 1 = 2{,}097.62\$.$$

Se questi token non fossero stati investiti all'interno del pool il valore posseduto dagli LP sarebbe stato pari a:

$$2 \cdot 550 + 1{,}000 \cdot 1 = 2{,}100\$.$$

Da questi dati si può facilmente calcolare l'IL utilizzando la formula 3.15:

$$1 - \frac{2{,}097.62}{2{,}100} = 0.0011,$$

che in percentuale corrisponde ad una perdita, del guadagno potenziale, dello 0.11%.
La formula 3.15 può essere facilmente generalizzata per pool contenti più token o contenuti in proporzioni differenti:

$$IL = \frac{Ratio_{invariant}}{Ratio_{hold}} \tag{3.16}$$

Dove $Ratio_{invariant}$ è il rapporto tra il valore posseduto nel pool e l'invariante k:

$$Ratio_{invariant} = \frac{V_{POOL}}{k} = \frac{\prod_{i=1}^{n}(x_i \cdot \Delta x_i)^{w_i}}{k} \tag{3.17}$$

Δx_i rappresenta la variazione percentuale del valore del token.
$Ratio_{hold}$ indica invece la variazione che si sarebbe verificata con il semplice holding degli asset ed è espressa come:

$$Ratio_{hold} = \sum_{i=1}^{n}(\Delta x_i \cdot w_i) \tag{3.18}$$

I grafici che si possono tracciare utilizzando queste formule sono riportati in figura 3.7.

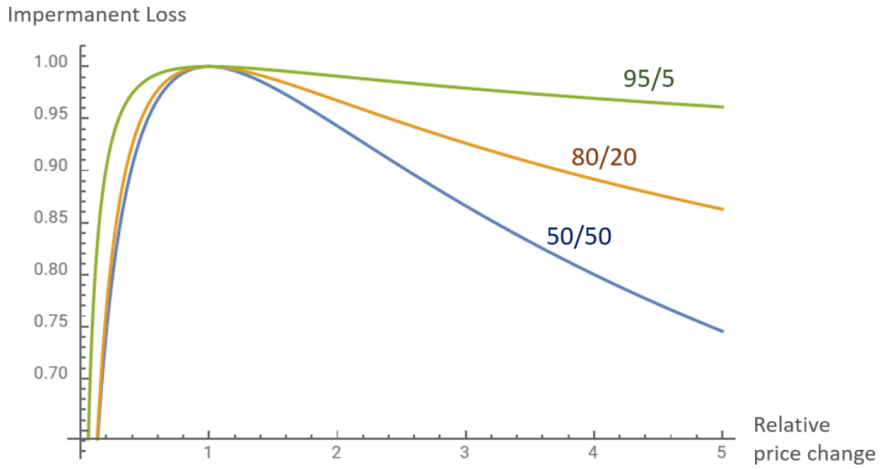


Figura 3.7. Grafico dell'IL per diverse pool [6]

Risulta chiaro come la scelta di un pool sbilanciato possa essere utile per mitigare l'effetto

[6] Fonte: https://docs-v2.balancer.fi/concepts/advanced/impermanent-loss.html.

dell'IL.
Negli AMM a liquidità concentrata l'IL viene formulata in modo differente perché non sarà influenzata solamente della volatilità, ma anche dall'intervallo scelto dall'LP.
Per rendere più chiara la lettura della formula verranno riportati V_{POOL} e V_{HOLD} separatamente. I passaggi seguiti per ottenere queste quantità sono analoghi a quelli della dimostrazione 3.2.1. Considerando sempre p come il rapporto tra p_1 e p_0 si possono definire le due quantità:

$$V_{\text{POOL}} = p_1 \cdot x_1 + y_1 = \begin{cases} \tilde{L} \cdot p_1 \left(\frac{1}{P_{min}} - \frac{1}{P_{max}} \right) & \text{se } p_1 < P_{min} \\ \tilde{L} \left(2p_1 - P_{min} - \frac{p_1}{P_{max}} \right) & \text{se } P_{min} \leq p_1 < P_{max} \\ \tilde{L} \cdot (P_{max} - P_{min}) & \text{se } p_1 \geq P_{max} \end{cases} \tag{3.19}$$

$$V_{\text{HOLD}} = p_1 \cdot x_0 + y_0 = \begin{cases} \tilde{L} \cdot p_1 \left(\frac{1}{P_{min}} - \frac{1}{P_{max}} \right) & \text{se } p_0 < P_{min} \\ \tilde{L} \left(p_0 + p - P_{min} - \frac{p_1}{P_{max}} \right) & \text{se } P_{min} \leq p_0 < P_{max} \\ \tilde{L} \cdot (P_{max} - P_{min}) & \text{se } p_0 \geq P_{max} \end{cases} \tag{3.20}$$

Seguendo la formula 3.15, si osserva che anche in questo caso per ottenere il valore finale dell'IL utilizzare queste quantità con la stessa relazione:

$$1 - \frac{V_{\text{POOL}}}{V_{\text{HOLD}}}.$$

Heimbach et al. [2022] analizzano come l'IL vari con intervalli di riferimento diversi. Viene

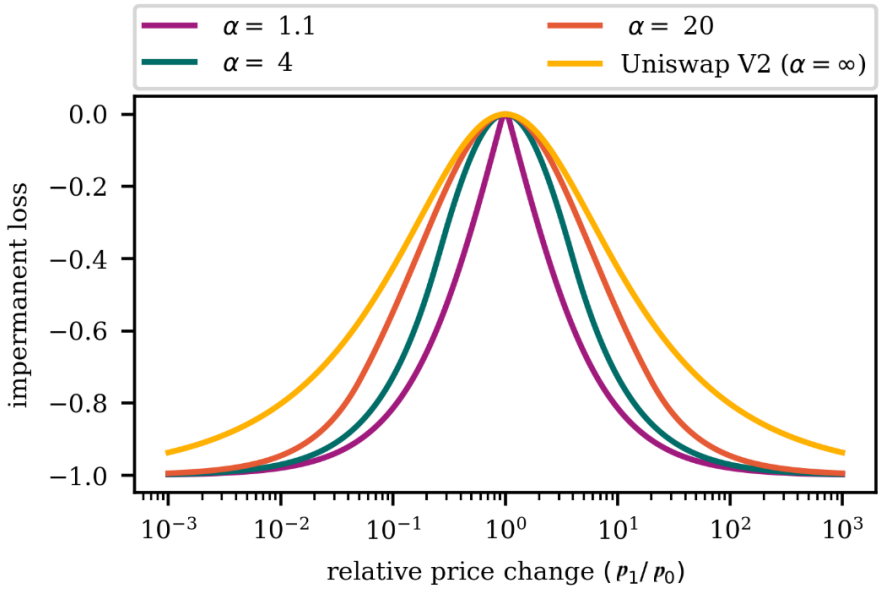


Figura 3.8. Diversa IL per intervalli di investimento [Heimbach et al., 2022]

preso in considerazione l'intervallo:

$$\left[\frac{1}{\alpha}p_0,\ \alpha p_0\right]$$

con $\alpha \in [1.1, 4, 20]$. Da questo studio emerge che più piccolo è α, ovvero più stretto è l'intervallo di prezzo, più rapidamente aumenta l'IL. Avendo come conseguenza, dunque, non solo un rischio maggiore che la liquidità diventi inattiva, ma anche una maggiore esposizione alla perdita impermanente. Questo è evidente dal grafico in figura 3.8.

Capitolo 4

Obiettivi e metodologia

L'obiettivo che si vuole raggiungere in questo elaborato è rispondere a due principali domande di ricerca:

- Quali sono i principi e i passaggi fondamentali per progettare un liquidity pool che garantisca sostenibilità economica nel lungo periodo?
- Quali strategie e meccanismi di incentivazione possono essere adottati per attrarre e mantenere gli LP all'interno di un liquidity pool?

Al fine di affrontare questi quesiti è stato impostato il processo di ricerca, volto ad analizzare i meccanismi di funzionamento dei liquidity pool e le dinamiche che influenzano la partecipazione degli investitori.
Questo processo è fondamentale per comprendere le diverse decisioni che sono necessarie al fine di una progettazione consapevole e razionale. Ciò perché ogni caratteristica scelta per il proprio pool influenzerà direttamente il suo funzionamento e il livello di partecipazione dei fornitori di liquidità. Nel paragrafo 4.1, infatti, viene analizzato dettagliatamente il processo decisionale da seguire in questa fase, prima dell'effettiva implementazione. Introducendo, inoltre, anche i possibili incentivi che possono essere distribuiti agli LP per compensare le potenziali perdite dovute all'IL.
La scelta del tipo di AMM, della coppia di token e del capitale da investire inizialmente risultano determinanti per la sostenibilità e l'efficienza del pool. Gli agenti economici che vogliono intraprendere processi di digitalizzazione per mezzo di tecnologie decentralizzate possono così influenzare la liquidità disponibile inizialmente, il livello di slippage atteso, cercando comunque di garantire un equilibrio tra rischio ed incentivi.
Si ha anche l'obiettivo di fornire uno strumento pratico allo sviluppatore di un nuovo liquidity pool per definire con maggiore precisione i parametri da utilizzare durante lo sviluppo concreto. A tal fine nella sezione 4.2 viene definita la funzione di utilità relativa al proprietario del pool, che assumerà il ruolo di funzione obiettivo all'interno del modello di ottimizzazione. Questa funzione ha lo scopo di modellare profitti e investimenti di chi lancia il pool, ma anche l'impatto sulle dinamiche degli LP, tenendo conto del loro ruolo e degli incentivi necessari per garantirne la partecipazione. Vengono, infatti, introdotte diverse strategie di incentivazione, come meccanismi di protezione dall'IL o distribuzione

di ricompense per il lockup. L'importanza di questo aspetto viene approfondita in seguito attraverso la spiegazione dettagliata della logica del modello in 4.2.
Nella definizione di tale funzione è stato necessario introdurre esplicitamente la liquidità, poiché ha un ruolo fondamentale nello stabilire i legami che intercorrono tra le variabili di interesse. Per raggiungere questo scopo sono stati utilizzati modelli di regressione applicati a dati reali, in modo da individuare legami chiave con il volume delle transazioni e la deviazione standard dei prezzi. Con questo approccio si può ottenere una stima più accurata e realistica dell'evoluzione della liquidità nel tempo, così che il modello sia in grado di ottimizzare i parametri in funzione delle condizioni di mercato.
Il modello, una volta definito, viene testato su scenari ipotetici per valutarne l'efficacia e tali risultati vengono trattati nel capitolo 5.

4.1 Processo decisionale

La creazione di un liquidity pool, come sottolineato in precedenza, richiede una valutazione accurata di aspetti fondamentali per ottenere un risultato dall'effettivo impatto strategico. Le decisioni prese, dalla scelta della coppia token, alla selezione dell'AMM, influenzano direttamente la stabilità e l'efficienza del pool, determinandone l'attrattività per gli LP.
Per tale motivo è importante seguire un processo decisionale ben strutturato, così da bilanciare i rischi e le opportunità. Si analizzano, dunque, di seguito i principali passi da seguire per avviare un liquidity pool, fornendo un quadro chiaro di tutte le variabili da prendere in considerazione.
Il primo passo nella configurazione di un liquidity pool riguarda la selezione della coppia di token che verrà utilizzata. Esistono diverse combinazioni possibili, di seguito vengono analizzate le seguenti:

- due stablecoin;
- due token di apprezzamento;
- una stablecoin e un token di apprezzamento;
- due token correlati positivamente.

Questa scelta incide direttamente sulla volatilità del pool, sulla profondità della liquidità e sul rischio per gli LP.
Ognuna di queste coppie presenta, infatti, caratteristiche differenti. Nel primo caso ci si inserisce in un mercato relativamente stabile, in cui i prezzi subiscono raramente variazioni notevoli, riducendo al minimo il rischio di IL. Questo tipo di pool è adatta nel caso in cui si voglia minimizzare la volatilità e attrarre investitori con un'alta avversione al rischio. Al contrario i volumi di scambio saranno inferiori poiché tale stabilità non porta ad avere opportunità di arbitraggio sfruttabili dai trader.
Per assecondare una maggiore propensione al rischio può essere, invece, scelta la coppia composta da due token di apprezzamento. Questi ultimi sono soggetti a variazioni di prezzo significative, offrendo interessanti opportunità di guadagno, grazie ai grandi volumi di scambi generati dagli arbitraggi. Il principale svantaggio di questa combinazione,

tuttavia, risulta essere l'elevata esposizione all'IL, che può incidere negativamente sui rendimenti degli LP.
Un equilibrio tra i due casi precedenti è quello in cui la coppia è costituita da una stablecoin ed un token di apprezzamento. La prima, infatti, grazie alla sua stabilità riduce la volatilità complessiva del pool, che dipenderà solo dal token di apprezzamento. Le variazioni di quest'ultimo possono, però, garantire volumi di scambio sufficienti ad assicurare un guadagno agli LP, che saranno comunque esposti all'IL.
Un ulteriore vantaggio che si può trarre scegliendo questa combinazione per il proprio pool è che il prezzo del token sarà determinato esclusivamente da domanda e offerta dello stesso. I ribilanciamenti fatti dall'AMM, infatti, non saranno tendenzialmente implicabili alla stablecoin, dimostrando che le variazioni di valore del token saranno esclusivamente attribuibili alle fluttuazioni di mercato.
L'ultima coppia può essere vista come un caso particolare della combinazione di due token di apprezzamento, a condizione che essi siano correlati positivamente. Viene, tuttavia, trattato separatamente poiché comporta implicazioni diverse, influenzando in modo distinto le dinamiche del pool e le strategie di gestione della liquidità. La correlazione positiva porta i prezzi dei due token, con buona probabilità, a muoversi nella stessa direzione, riducendo il rischio di squilibri e di conseguenza di IL. In questo caso, però, essendo comunque soggetti alla volatilità, ci si possono aspettare guadagni maggiori rispetto alla combinazione di due stablecoin. Il rischio che si corre in questo caso è una maggiore esposizione a eventi di mercato che possono influenzare entrambi i token in modo simultaneo. Se entrambi gli asset appartengono, infatti, allo stesso ecosistema, potrebbero essere colpiti contemporaneamente un evento negativo, amplificando le perdite per gli LP.
Le caratteristiche di ogni coppia sono state riassunte nella tabella 4.1.

Coppia di token	**Pro**	**Contro**
Due stablecoin	Basso rischio di IL	Volumi inferiori per assenza di arbitraggio
Due token di apprezzamento	Volumi di scambio elevati	Rischio elevato di IL
Una stablecoin e un token di apprezzamento	Prezzo determinato da domanda e offerta	Rischio di IL
Due token correlati positivamente	Minore rischio di IL	Sensibili ad eventi di mercato negativi

Tabella 4.1. Pro e contro per ogni coppia di token

Nel caso in cui uno dei due token sia nuovo sul mercato è importante definirne il prezzo iniziale in modo accurato. Esistono due possibili soluzioni: la prima consiste nel confrontare il proprio token con quelli già esistenti, con caratteristiche analoghe, e si attribuisce

il valore per similitudine; la seconda è quella di implementare un bootstrap AMM, come descritto nel paragrafo 3.1.6. Solitamente, se non si sceglie quest'ultimo metodo, il prezzo che viene assegnato inizialmente tende a essere relativamente basso per attrarre maggiore liquidità, rischiando però acquisti in grandi quantità da parte dei bot. Essi potrebbero sfruttare il prezzo sottostimato per ottenere un vantaggio immediato, alterando la distribuzione iniziale del token e potenzialmente destabilizzando il pool, come precedentemente descritto nella sezione 3.1.6.
Una volta stabilita la coppia di token, il passo successivo consiste nella scelta dell'AMM più adatto alle caratteristiche della coppia selezionata e agli obiettivi del pool. Per ognuna di esse vengono descritti vantaggi e svantaggi relativi ai diversi AMM introdotti nel paragrafo 3.1.
Prendendo in considerazione la combinazione di due stablecoin l'AMM più indicato risulta essere il modello ibrido, essendo stato progettato appositamente per questo tipo di coppie. Favorisce, infatti, la stabilità seppur non permetta di ottenere rendimenti molto elevati. Per favorire questi ultimi, infatti, si potrebbe optare per un AMM a liquidità concentrata, a discapito, però, della stabilità che risulta essere inferiore. In questo caso, inoltre, risulterebbe poco vantaggioso implementare un CPMM O CMMM, poiché poco efficienti con questo tipo di asset.
La situazione cambia significativamente per le restanti tre coppie, per le quali l'AMM a media costante risulta essere spesso un'ottima scelta. Questo modello offre una gestione più equilibrata della liquidità, risultando particolarmente utile per coppie con dinamiche di prezzo più variabili rispetto alle stablecoin, permettendo di mitigare l'effetto dell'IL. La definizione corretta dei pesi, tuttavia, può risultare complessa e richiedere un'analisi accurata. Al contrario l'AMM ibrido non è considerato un'alternativa adeguata a queste coppie, poiché l'elevata volatilità potrebbe favorire lo svuotamento dei pool, in seguito ad operazioni di arbitraggio.
Si può osservare, inoltre, che nel caso in cui il pool sia composto da due token di apprezzamento potrebbe risultare vantaggiosa anche l'implementazione di un AMM a liquidità concentrata. Così sarebbe possibile ottenere rendimenti elevati, migliorando l'efficienza del capitale, a patto che si abbia una gestione attiva del portafoglio.
Il CPMM, infine, può essere implementato nei pool con token correlati positivamente, offrendo un meccanismo di scambio efficiente in queste configurazioni. Grazie alla correlazione tra gli asset è possibile contenere l'IL, che risulterebbe più significativa se si implementasse questo AMM per le altre combinazioni.
Nei due scenari in cui la scelta dell'AMM a liquidità concentrata può essere vantaggiosa, possono essere fornite indicazioni utili per gli intervalli di investimento. Tale informazione può aiutare lo sviluppatore a indirizzare i futuri fornitori di liquidità nella selezione dei range di prezzo più adatti, migliorando così l'efficienza del pool. Nel caso di una coppia composta da due stablecoin, l'approccio più efficace è quello di adottare un range di prezzo molto stretto, poiché la loro volatilità è estremamente ridotta e le oscillazioni di prezzo minime. Questo consente di massimizzare l'efficienza del capitale investito e ridurre lo slippage nelle operazioni di scambio. Al contrario, nel caso di una coppia formata da due token di apprezzamento, è preferibile un range più ampio, poiché questi asset sono soggetti a fluttuazioni di prezzo più marcate. Un intervallo troppo stretto potrebbe, infatti, richiedere frequenti ribilanciamenti, aumentando il rischio di fuoriuscita della liquidità

dall'intervallo definito e riducendo l'efficienza complessiva del pool.
Una volta prese queste decisioni è necessario determinare il valore dell'investimento in fase iniziale. Per questo motivo è utile definire accuratamente come influisce lo slippage in relazione agli asset presenti nel pool. Se si assume la stessa profondità di liquidità per ogni scenario, si può osservare che lo slippage varia in base alla tipologia di asset scelto. In particolare, il minore slippage si verifica nel caso delle stablecoin, il massimo slippage si registra invece, nei pool composti da token di apprezzamento. Questo risultato è attribuibile direttamente alle differenze di volatilità tra i token, che determinano il grado di oscillazione dei prezzi all'interno del pool.
Quando il pool è composto da una stablecoin e un token di apprezzamento, oppure da due token correlati positivamente, lo slippage si posiziona tra i due estremi precedenti. Nel primo caso, la stablecoin attenua le variazioni di prezzo, ma non elimina l'impatto della volatilità del token di apprezzamento, nel secondo, la correlazione tra i due asset riduce il rischio di squilibri improvvisi, ma, nonostante ciò, in entrambi i casi lo slippage risulta comunque superiore a quello di una coppia di sole stablecoin.
Queste osservazioni permettono di conseguenza di definire il livello ottimale di liquidità da fornire al pool, in modo da ridurre lo slippage. Nei pool con stablecoin, infatti, può essere sufficiente un livello di liquidità inferiore rispetto ai pool composti da token di apprezzamento, che richiedono invece una maggiore profondità per mitigare le fluttuazioni di prezzo. Nei casi rimanenti, dunque, come quelli con token correlati o una coppia composta da una stablecoin e un token di apprezzamento, la quantità di liquidità necessaria è inferiore rispetto ai pool con due token di apprezzamento.
Dopo aver definito la struttura del pool è essenziale ottimizzare l'uso del capitale e mitigare il rischio di IL, in quando sono aspetti fondamentali per assicurare l'attrattività del pool. Per soddisfare questi requisiti, infatti, la scelta dell'AMM può essere ulteriormente affinata: per migliorare l'efficienza del capitale, una strategia efficace è l'adozione di un AMM con liquidità concentrata, mentre per ridurre l'IL si può optare per un CMMM. Se invece l'obiettivo è ridurre l'IL, è possibile implementare un CMMM. Se si dispone di ulteriore capitale, inoltre, si possono offrire incentivi agli LP, aumentando così la partecipazione e la stabilità del pool, questo aspetto viene approfondito nel prossimo paragrafo 4.1.1.
In sintesi, i passi da seguire sono i seguenti:

1. selezione della coppia di token;

2. scelta del tipo di AMM;

3. definizione del livello di liquidità;

4. ottimizzazione del capitale e mitigazione dell'IL.

Tale processo è riportato in modo schematico nel diagramma di flusso riportato nella figura 4.1. Seguendo questo flow chart, è possibile avere una guida per individuare le caratteristiche ottimali, garantendo così una configurazione del pool coerente con i propri obiettivi.

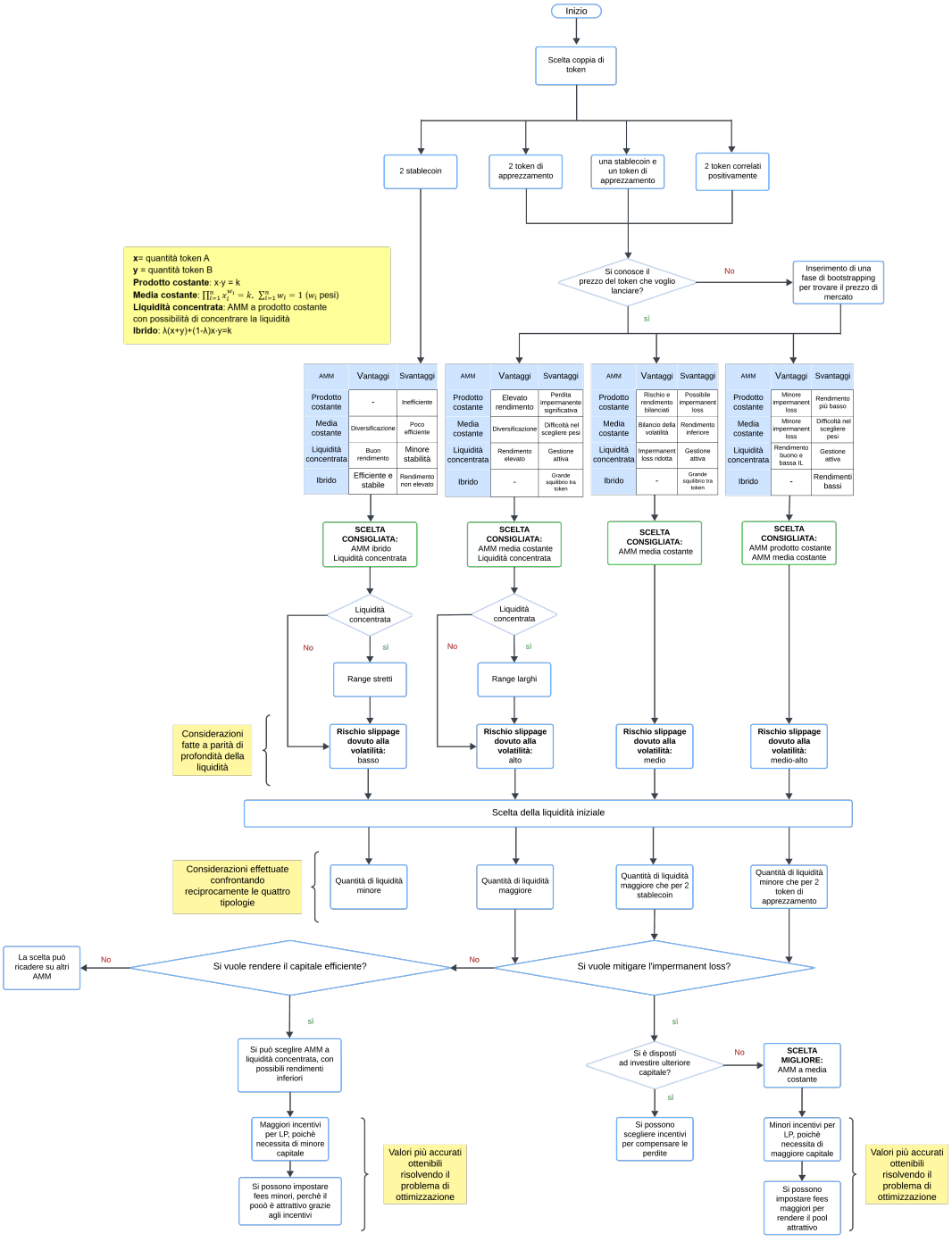


Figura 4.1. Diagramma di flusso relativo al setup

4.1.1 Incentivi

Una buona struttura del pool, però, non è sufficiente per attrarre gli LP: è fondamentale, infatti, implementare meccanismi di incentivazione che lo rendano più competitivo e sostenibile nel lungo periodo.
Esistono diverse strategie che possono essere adottate, in questo paragrafo verranno trattate le principali:

- incentivi per gli early adopter;
- incentivi per il completamento di un periodo di lockup;
- distribuzione di LP token;
- distribuzione di token di governance;
- assicurazioni contro l'IL.

Gli incentivi per gli early adopter hanno l'obiettivo di premiare i primi fornitori di liquidità, offrendo bonus iniziali che si aggiungono alle commissioni generate dagli scambi nel pool. Questa strategia è particolarmente utile per attrarre liquidità nelle fasi iniziali e garantire che il pool raggiunga una profondità sufficiente per funzionare in modo efficiente. Questo meccanismo risulta particolarmente efficace nei pool creati per il lancio di un nuovo token, dove è necessario incentivare gli LP a fornire liquidità fin dalle prime fasi. Offrire ricompense iniziali permette di accelerare la crescita del pool e di rendere più fluido il mercato del nuovo asset, facilitandone l'adozione. Tali incentivi saranno maggiori nelle fasi iniziali per stimolare la partecipazione e la formazione della liquidità, per poi diminuire progressivamente nel tempo, una volta che il pool avrà raggiunto una stabilità sufficiente e non necessiterà più di incentivi elevati per attrarre fornitori di liquidità.
Può considerarsi opposto il caso in cui viene distribuito un incentivo a seguito del completamento di un periodo di lockup. In questo caso gli LP vengono ricompensati dopo aver mantenuto la loro liquidità nel pool per un periodo prestabilito, incentivandoli a non ritirare i propri fondi prematuramente. L'obiettivo è quello di garantire una maggiore stabilità e profondità del pool. A differenza degli incentivi per gli early adopter, che diminuiscono nel tempo, le ricompense per il lockup, invece, tendono ad aumentare progressivamente, premiando maggiormente gli utenti che vincolano la loro liquidità per periodi più lunghi. Questa strategia è particolarmente utile nei pool con token molto liquidi, dove può essere necessario contrastare l'uscita rapida di liquidità.
Un altro metodo che può essere utilizzato consiste nella distribuzione di token, che possono essere LP token o token di governance. I primi fungono da ricevuta della liquidità fornita e rappresentano la quota di ogni LP all'interno del pool. Oltre a ciò, possono essere utilizzati come collaterale in altri protocolli DeFi, moltiplicando così le opportunità di rendimento per gli LP. Questa caratteristica li rende adatti a qualsiasi tipo di pool, offrendo un ulteriore incentivo agli LP per mantenere i propri fondi vincolati nel protocollo. I secondi, invece, consentono agli LP di partecipare attivamente alle decisioni sul futuro del protocollo, rafforzando il loro coinvolgimento e incentivando una gestione più decentralizzata del pool. Grazie a questi token, gli utenti possono votare su aggiornamenti

del protocollo, modifiche delle fee o strategie di incentivazione, contribuendo direttamente allo sviluppo dell'ecosistema. Questa soluzione è particolarmente indicata per i pool più decentralizzati, in cui la governance distribuita rappresenta un elemento chiave per la sostenibilità del progetto.

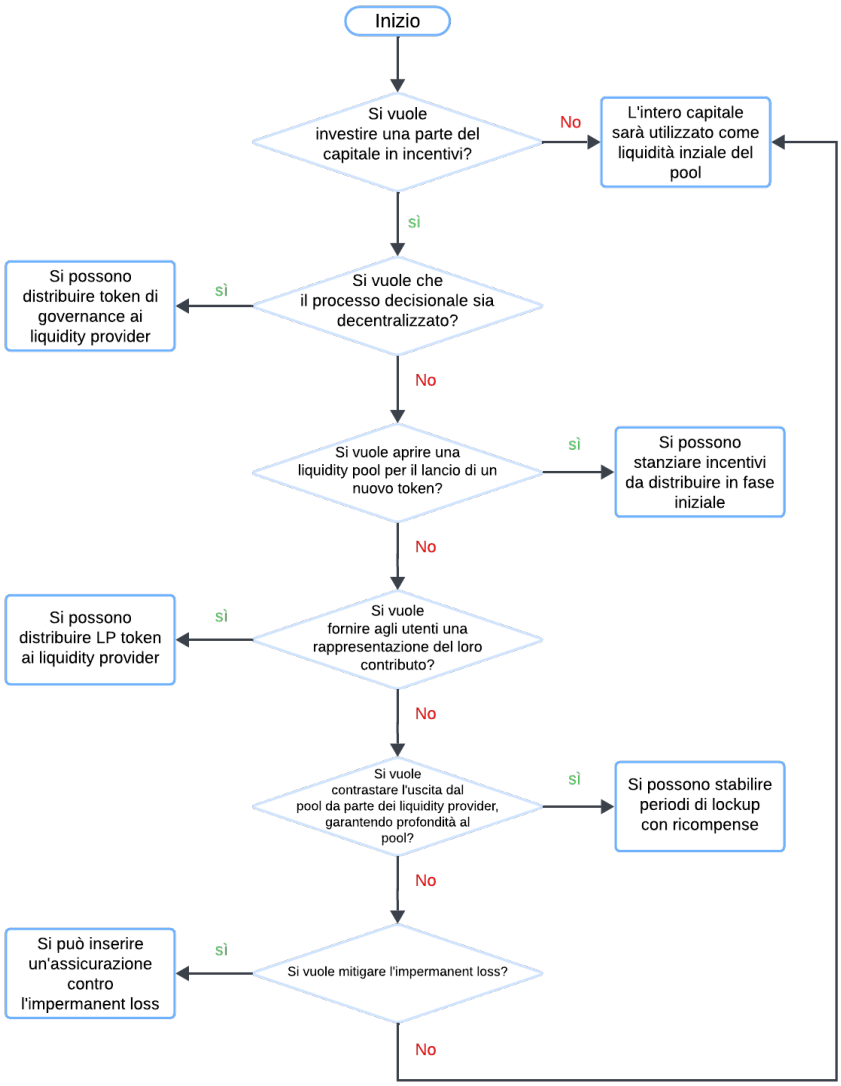


Figura 4.2. Diagramma di flusso relativo agli incentivi

L'ultimo caso è rappresentato dalle assicurazioni che permettono di ridurre i rischi associati alla partecipazione al pool. L'obiettivo è quello di compensare le perdite subite dagli LP, restituendo loro una percentuale dell'IL in base al periodo in cui la liquidità è rimasta vincolata nel pool. Il meccanismo prevede che il rimborso si accumuli giorno dopo giorno,

partendo da un minimo di 30 giorni di blocco e arrivando fino a una copertura massima del 100% delle perdite. Questa strategia risulta particolarmente utile nei pool con token molto volatili, dove il rischio di perdita impermanente è più elevato.
Anche in questo caso è stato fornito un diagramma di flusso, riportato in figura 4.2, che permette di scegliere la forma di incentivo migliore per il progetto da implementare. In questo caso si prevede l'implementazione di un solo tipo di incentivo per pool, ma tale ragionamento può essere generalizzato per inserirne diversi.
Le strategie di incentivazione devono essere, dunque, attentamente bilanciate in base agli obiettivi del liquidity pool. Mentre gli early adopter possono beneficiare di incentivi per il lancio di un nuovo token, gli LP a lungo termine possono essere premiati con token di governance, LP token, assicurazioni e periodi di lockup. Il diagramma fornisce una guida chiara per selezionare le opzioni più adatte, permettendo di creare un ecosistema sostenibile e attraente per gli investitori.
Si evidenzia che, seppur non vengano trattate in questo elaborato, possono essere utilizzate strategie di protezione da parte degli LP. Strategie di hedging possono essere applicate come nella finanza tradizionale, sfruttando derivati come opzioni, futures e swap.
Queste strategie, tuttavia, non vengono approfondite ulteriormente, in quanto riguardano esclusivamente la gestione diretta degli LP e non possono essere influenzate direttamente dal design e dalla struttura del liquidity pool.

4.2 Modello di ottimizzazione

Nel contesto del setup dei liquidity pool, la gestione ottimale delle caratteristiche è fondamentale per garantirne sia l'efficienza operativa che la stabilità del sistema finanziario decentralizzato. A tal fine, come già introdotto in precedenza, è stato sviluppato un modello di ottimizzazione che supporta gli sviluppatori nelle decisioni strategiche, fornendo una visione completa delle variabili coinvolte. Questa applicazione, implementata successivamente al processo decisionale descritto in 4.1, si configura come un ulteriore strumento in grado di tradurre in soluzioni operative le scelte stabilite in precedenza. Ciò contribuisce in modo determinante al perfezionamento e alla robustezza complessiva dei liquidity pool.
Tale modello risulta determinante, infatti, per ottenere soluzioni numeriche adatte al contesto di sviluppo. È possibile, così, confrontare come diversi fattori, quali il volume, la liquidità, l'IL e gli incentivi, interagiscano e forniscano soluzioni differenti. Lo sviluppatore, in questo modo, può analizzare come le diverse variabili impattano sul proprio pool.
Per definire matematicamente il modello è necessario introdurre il concetto di **utilità**. Quest'ultimo si utilizza, solitamente, per rappresentare il benessere che un individuo ottiene dal consumo di beni e servizi o dalle scelte di investimento, per poi essere tradotto in funzioni matematiche che ne quantificano il valore.
In finanza, infatti, di norma rappresenta un equilibrio tra rischio e rendimento, fungendo da criterio decisionale che aiuta gli investitori a individuare le soluzioni ottimali per la composizione del proprio portafoglio.
Nel contesto dell'ottimizzazione, dunque, l'utilità è spesso utilizzata come funzione obiettivo da massimizzare, permettendo di valutare diverse soluzioni. In questo elaborato viene

adottato lo stesso approccio: la funzione di utilità guida il processo decisionale, orientando la scelta verso la configurazione più vantaggiosa del pool.
In particolare, quella che si vuole definire è l'**utilità complessiva dello sviluppatore** come misura in grado di tenere in considerazione sia il proprio benessere che quello degli LP. Ciò perché, sebbene chi decida di intraprendere questo percorso voglia perseguire i propri obiettivi, il successo del pool dipende anche dalla partecipazione attiva degli LP. In questo modo, infatti, ci si assicura che le decisioni strategiche siano relative all'interesse collettivo, creando un equilibrio tra tutte le parti coinvolte. L'utilità dello sviluppatore considera, inoltre, i guadagni positivamente, incrementando il valore complessivo, mentre i costi $C(t)$ vengono considerati negativamente, contribuendo a ridurne il livello. In particolare, i primi derivano dalle commissioni ricevute, calcolate come il prodotto tra livello di fee f e il volume di trading al tempo t $\mathrm{V}(t)$, i secondi dalla somma di liquidità investita $L(T)$ e dagli incentivi erogati $I(T)$.
Per semplificare l'espressione del modello, si è scelto di aggregare il comportamento di tutti i fornitori di liquidità in un'unica funzione. In questo modo, il contributo complessivo degli LP viene considerato come un effetto collettivo sul pool, senza doversi occupare delle specificità individuali. L'utilità degli LP, analogamente, è definita come la differenza tra guadagni e perdite. In particolare, i guadagni sono costituiti dalle fee per transazione e dagli incentivi ricevuti, le perdite, invece, sono rappresentate da una stima dell'IL $P(t)$. Questi concetti vengono tradotti con le seguenti formule:

$$\mathrm{U_{SV}}(t) = (1-\beta)\cdot f \cdot V(t) - C(t) + \rho \cdot \mathrm{U_{LP}}(t) \tag{4.1}$$

$$\mathrm{U_{LP}}(t) = \beta \cdot f \cdot V(t) + I(t) - P(t) \tag{4.2}$$

$\mathrm{U_{SV}}(t)$ e $\mathrm{U_{LP}}(t)$ rappresentano rispettivamente l'utilità dello sviluppatore e quella dei LP. β è un parametro che assume valori in $[0,1]$ e definisce la ripartizione delle commissioni tra i proprietari del pool e i fornitori di liquidità, ρ, invece, descrive l'importanza che si vuole attribuire agli LP. Seppur possa essere fissato in fase di scrittura del modello, un'interpretazione coerente è quella di considerarlo come un parametro dinamico, che cresce nel tempo, affinché i LP assumano un ruolo sempre più centrale nelle fasi avanzate della vita del pool.
Questa combinazione evidenzia come il concetto di utilità sia trasversale, influenzando sia il comportamento dei consumatori che quello degli investitori. Integrando questo approccio è possibile individuare soluzioni numeriche ottimali che bilanciano la redditività e il rischio, garantendo un sistema finanziario decentralizzato più efficiente e stabile.
Si osserva che con tale formulazione si vuole sottolineare che i profitti sono direttamente proporzionali sia al volume di scambio sia al livello fee per transazione. L'IL, invece, avrà un impatto maggiore in presenza di variazioni di prezzo più accentuate, in quanto proporzionale alla variazione del rapporto tra il valore token del pool.
Questi ragionamenti costituiscono la base per introdurre la funzione obiettivo del modello di ottimizzazione in un'ottica dinamica:

$$\max\Big\{(1-\beta)\,f\cdot V(t) - \Big(I(t)+L(t)\Big) + \rho\Big[\beta\cdot f\cdot V(t) + I(t) - P(t)\Big] + \gamma\,\mathbb{E}\big[U_{SV}(t+1)\big]\Big\} \tag{4.3}$$

Il parametro γ rappresenta il fattore di decadimento e consente di regolare il peso attribuito al futuro nel modello. In fase di scrittura, γ viene fissato per decidere se dare maggiore

o minore importanza al valore atteso dell'utilità futura: un valore elevato comporta una forte considerazione dei benefici futuri, mentre un valore ridotto evidenzia una preferenza per i risultati immediati.
Viene utilizzato questo approccio per garantire che le decisioni siano adatte all'evoluzione del pool nel lungo periodo con una visione che non consideri esclusivamente l'istante iniziale.
L'obiettivo del modello, con tale funzione obiettivo, è quello di individuare la ripartizione ottimale del capitale tra incentivi e liquidità, al fine di massimizzare l'efficienza del pool. Successivamente, si concentra sull'analisi dinamica della distribuzione degli incentivi, istante per istante, per garantire un'allocazione efficace delle risorse nel tempo.
Definite queste proprietà del modello possono essere introdotti i vincoli di base che devono essere rispettati per l'ottimizzazione:

$$\begin{aligned} M &= I_{tot} + L_{tot} \\ \sum_{t=0}^{T} I(t) &= I_{tot} \end{aligned} \tag{4.4}$$

Dove M rappresenta il capitale iniziale, I_{tot} e L_{tot} sono rispettivamente gli incentivi totali e la liquidità totale.
Può essere introdotta, inoltre, la possibilità di reinvestire le fee raccolte in incentivi, incrementando così una partecipazione più attiva da parte degli LP, ottenendo un ulteriore vincolo:

$$W(t) = I(t) + \tau \cdot [(1 - \beta) \cdot f \cdot V(t-1)] \tag{4.5}$$

Con $W(t)$ si esprime la nuova quantità di incentivi, da sostituire a $I(t)$ in 4.2, con τ, parametro stabilito a priori, si indica la porzione di commissioni che vuole essere reinvestita.
È importante anche garantire che la liquidità sia sufficientemente elevata da far fronte allo slippage. Prendendo in considerazione un CPMM è facile dimostrare che una transazione pari al 10% della liquidità causerebbe uno slippage circa del 9%. Per tale motivo è utile introdurre un vincolo sulla liquidità minima, che deve essere pari ad almeno 100 volte il valore di una transazione media.

$$L_{tot} \geq 100 \cdot T_{media} \tag{4.6}$$

Dove con T_{media} si indica il valore medio di una transazione. Per una maggiore stabilità, questo rapporto può essere esteso fino a 1000 volte.
Questi vincoli costituiscono le basi necessarie per l'ottimizzazione, assicurando che il modello operi in un equilibrio che valorizzi sia gli obiettivi dello sviluppatore sia quelli degli LP.

Considerando tali vincoli si ottiene:

$$\max\Big\{(1-\beta)\, f \cdot V(t) - \Big(W(t) + L(t)\Big) + \rho\Big[\beta \cdot f \cdot V(t) + W(t) - P(t)\Big] + \gamma\, \mathbb{E}\big[U_{SV}(t+1)\big]\Big\}$$

s.t.:

$$M = I_{tot} + L_{tot}$$

$$\sum_{t=0}^{T} I(t) = I_{tot}$$

$$W(t) = I(t) + \tau \cdot [(1-\beta) \cdot f \cdot V(t-1)]\,, \qquad \forall t \in [1, T]$$

$$L_{tot} \geq 100 \cdot T_{media}$$

In questo modello il volume viene considerato un fattore esogeno, al contrario la forma funzionale della liquidità deve essere determinata. Tale processo viene spiegato dettagliatamente nel prossimo paragrafo 4.2.1.

4.2.1 Forma funzionale liquidità

In questo paragrafo si affronta la ricerca della forma funzionale della liquidità, partendo dall'analisi di dati reali raccolti dai pool. Servendosi di questi ultimi, l'obiettivo iniziale è quello di identificare le relazioni che meglio descrivono il comportamento della liquidità. Per introdurre anche una dipendenza dagli incentivi, invece, viene utilizzato un approccio teorico, non basato su evidenze empiriche. Ciò è utile per ottenere una rappresentazione completa della liquidità.
A tal fine sono stati estratti i dati da due differenti liquidity pool per poter comprendere il tipo di relazione esistente e testarne la veridicità. In particolare, durante l'analisi, sono state raccolte informazioni relative ai livelli di liquidità e ai volumi di scambio su un periodo di circa quattro anni, per ottenere un quadro approfondito e a lungo termine. È stata condotta, inoltre, un'analisi per l'estrazione dei prezzi dei token, integrando così l'approfondimento delle dinamiche di mercato.
Per garantire l'accuratezza e la rappresentatività dei dati analizzati, la ricerca si è concentrata sui liquidity pool presenti su Uniswap v2, in quanto, come emerge dall'immagine 4.3, è la versione che registra maggior volume di scambi[1], in tale piattaforma. In particolare, sono stati esaminati due pool distinti: il primo, relativo alla coppia WETH/USDC, è stato selezionato in quanto tali token rappresentano quelli più utilizzati e liquidi[2], come si osserva dal grafico in figura 4.4, offrendo così un indicatore affidabile per l'analisi dei meccanismi di mercato. Il secondo, composto da WETH e WBTC, è stato scelto per osservare anche le interazioni in caso di due token di apprezzamento.
Per garantire la replicabilità del processo il codice utilizzato per l'estrazione della liquidità e dei prezzi è riportato nell'appendice A. Grazie a ciò è stato possibile calcolare la deviazione standard del rapporto tra il valore dei due token nel pool, utile per le ricerche

[1] Fonte: https://dune.com/KARTOD/Uniswap-Mega-Dashboard-V2.

[2] Fonte: https://dune.com/mbuffara/uniswap-pools#pools.

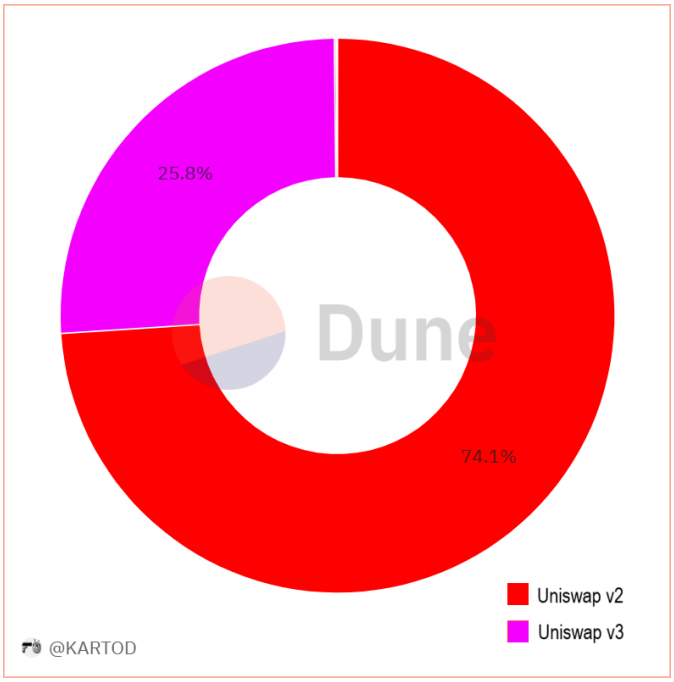


Figura 4.3. Suddivisione del trading sulla piattaforma Uniswap

svolte. I dati relativi al volume, invece, sono ottenuti eseguendo gli script in appendice B.
La raccolta di questi dati è stata fondamentale per ottenere le relazioni presenti tra le variabili. È stato, infatti, possibile svolgere un'approfondita analisi statistica che potesse confermare la veridicità di tali legami.
Per avviare tale percorso è stata calcolata la correlazione sia tra le variabili originali, ossia liquidità, volume e deviazione standard del rapporto dei prezzi, sia tra le loro trasformazioni logaritmiche, al fine di evidenziare eventuali relazioni non lineari. L'analisi mostra che le correlazioni risultano positive in entrambe i casi, ma assumono valori significativamente maggiori passando alle trasformazioni logaritmiche. In particolare, la relazione più evidente si riscontra tra le trasformazioni logaritmiche di liquidità e volume. In questo modo, è stato possibile stabilire una solida base per l'ulteriore formalizzazione della forma funzionale. I risultati dettagliati sono riportati nell'appendice D, l'immagine D.1 relativa ai dati originali, la D.2, invece, rappresenta quelli in logaritmo.
Per rendere più esplicite le relazioni individuate sono stati applicati modelli di regressione. Inizialmente si è concentrato lo studio su liquidità e volume, per poi integrare anche la deviazione standard del rapporto dei prezzi.
Si è potuto osservare che confrontando diversi tipi di relazioni, in particolare lineare, polinomiale e logaritmica, tra le prime due variabili, il modello con statistiche migliori è risultato essere il log-log.
La bontà di un modello di regressione viene spesso valutata tramite il **coefficiente di**

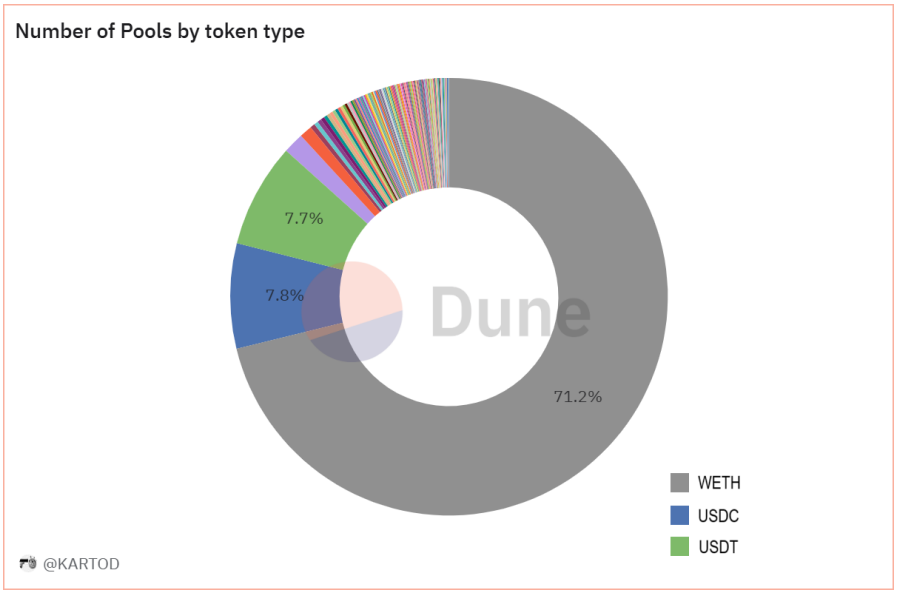


Figura 4.4. Frequenza dei token nelle pool di Uniswap

determinazione, definito come:

Definizione: 4.2.1.

$$R^2 = 1 - \frac{RSS}{TSS},$$

dove:
Devianza residua (Residual Sum of Squares, RSS):

$$RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 ,$$

Devianza totale (Total Sum of Squares, TSS):

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2 ,$$

con:

- $\hat{y}_i$ *che rappresenta i dati stimati dal modello,*
- y_i *i dati osservati,*
- $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$*, la media dei dati osservati.*

In alternativa può essere utilizzato il **criterio d'informazione di Akaike**, o Akaike's information criterion (AIC), definito come:

Definizione: 4.2.2.

$$AIC = 2k - 2\ln(L),$$

dove k è il numero di parametri del modello statistico e L rappresenta il valore massimizzato della funzione di verosimiglianza del modello stimato.

Il primo indice deve tendere a uno per indicare una buona capacità esplicativa del modello, il secondo, invece, deve essere il più basso possibile. Il modello che applica la trasformazione logaritmica sia al volume che alla liquidità risulta essere quello che soddisfa al meglio questi criteri.
Questo risulta evidente osservando la figura 4.5.

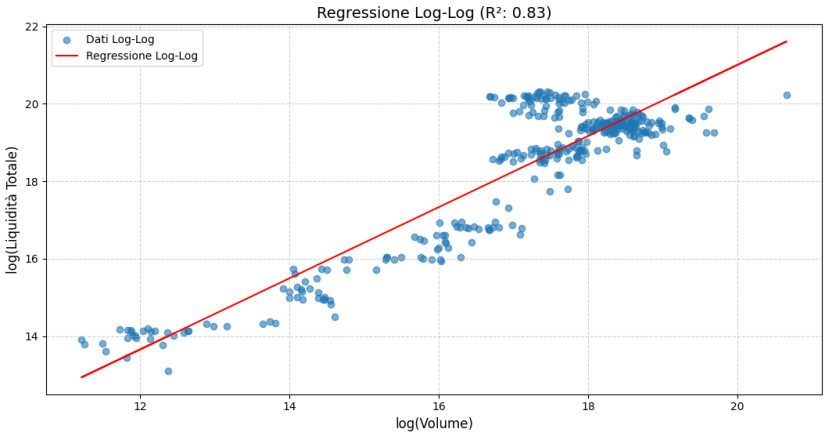


Figura 4.5. Regressione log-log — $R^2 = 0.83$

Ottenendo così una relazione del tipo:

$$log(L(t)) = \beta_0 + \beta_1 \cdot log(V(t)) \tag{4.7}$$

Dove β_0 rappresenta l'intercetta e $\beta_1 > 0$ il coefficiente relativo al volume, t tempo espresso in giorni. Quest'ultimo varrà per tutta la trattazione del modello.
È stato analogo il processo svolto con l'inserimento della deviazione standard, ottenendo anche in questo caso lo stesso tipo di relazione:

$$log(L(t)) = \beta_0 + \beta_1 \cdot log(V(t)) + \beta_2 \cdot log(\sigma(t)) \tag{4.8}$$

In questo caso però β_2, il coefficiente relativo alla deviazione standard, assume un valore negativo. Ciò può essere interpretato come un'indicazione che l'aumento della volatilità porta a una diminuzione della liquidità, in quanto l'incremento del rischio di IL tende a dissuadere gli LP dall'investire.
L'analisi delle statistiche ha evidenziato, dunque, che la relazione logaritmica riesce a legare in maniera efficace tutte e tre le variabili considerate.
Per proseguire è utile fornire la definizione di **p-value**:

Definizione: 4.2.3. *Sia H l'ipotesi che il valore x dei dati osservati sia estratto da una certa variabile aleatoria X nota. Il p-value è definito come la probabilità, supposta l'ipotesi H, di ottenere un risultato uguale o più estremo di quello effettivamente osservato. Il p-value è definito come:*

- $Pr(X \geq x \mid H)$ *per test unilaterali destri;*
- $Pr(X \leq x \mid H)$ *per test unilaterali sinistri;*
- $2\min\{Pr(X \leq x \mid H),\ Pr(X \geq x \mid H)\}$ *per test bilaterali.*

Un p-value piccolo implica una maggiore significatività del test, in quanto suggerisce che l'ipotesi H non riesce a spiegare adeguatamente i dati osservati. In altre parole, diventa sempre meno plausibile che il valore ottenuto sia stato effettivamente generato dalla variabile aleatoria X.
Ciò è importante per poter comprendere quali componenti autoregressive sono necessarie nel modello per evidenziare le dinamiche temporali. L'inclusione di questi termini consente di catturare le dipendenze tra le osservazioni nel tempo, rendendolo così ancora più robusto.
Svolgendo, dunque, un'analisi statistica approfondita, osservando i p-value relativi, si è osservato che le componenti logaritmiche a lag 1 per volume e liquidità sono statisticamente significativa, al contrario di quelle a lag 2. Con il termine *lag* si indica il numero di periodi di ritardo presi in considerazione: ad esempio, un modello con lag 1 tiene conto dell'influenza del valore osservato al tempo $t-1$ sulla variabile al tempo t. Questo risultato suggerisce che l'effetto della dipendenza temporale si manifesta principalmente in maniera immediata, rafforzando l'importanza delle componenti autoregressive nel migliorare la capacità predittiva del modello.
Questo approccio metodologico, che ha previsto prima l'analisi del volume e della deviazione standard in forma logaritmica e poi l'integrazione di componenti autoregressive, ha contribuito in modo determinante alla definizione della forma funzionale della liquidità e alla validazione dei risultati empirici.
Si ottiene così la seguente forma funzionale:

$$\log(L(t)) = \beta_0 + \beta_1 \cdot \log(V(t)) + \beta_2 \cdot \log(\sigma(t)) + \beta_3 \cdot \log(V(t-1)) + \beta_4 \cdot \log(L(t-1)) \quad (4.9)$$

O più esplicitamente:

$$L(t) = e^{\beta_0} \cdot V(t)^{\beta_1} \cdot \sigma(t)^{\beta_2} \cdot V(t-1)^{\beta_3} \cdot L(t-1)^{\beta_4} \quad (4.10)$$

L'approccio combinato, che integra l'analisi della correlazione con l'applicazione di modelli di regressione, ha quindi offerto una prospettiva metodologica robusta e completa, garantendo, non solo l'accuratezza dei risultati, ma anche la loro riproducibilità. Tutti i passi che sono stati introdotti per raggiungere tale forma funzionale, infatti, possono essere verificati utilizzando il codice in appendice C.
Come già introdotto in precedenza, per rendere l'equazione 4.10 più completa, è necessario integrare il contributo degli incentivi. L'obiettivo è quello di valutare come questi ultimi influenzino il comportamento dei fornitori di liquidità e di conseguenza della liquidità.

In questo caso, però, a causa della scarsità di dati empirici, non è stato possibile basare l'analisi su osservazioni reali rendendo fondamentale l'adozione di un approccio teorico.
Per questo motivo, è stata sviluppata una procedura di generazione di dati sintetici, ideata per replicare, in un ambiente controllato, le ipotesi formulate sulle relazioni tra le variabili. Questi dati artificiali sono stati ottenuti mediante modelli matematici e simulazioni, utilizzando parametri derivati da ipotesi teoriche, e hanno permesso di esaminare in dettaglio come variazioni negli incentivi possano influire sui comportamenti osservabili nel sistema. Tale analisi è stata svolta in relazione agli incentivi per gli early adopter e per ricompense successive ad un periodo di lockup.
In relazione al primo caso, dunque, analizzando le caratteristiche di tale incentivo riportate nel paragrafo 4.1.1, si è ipotizzato che vengano distribuiti seguendo questa funzione:

$$I(t) = I_{max} \cdot e^{-\epsilon t} \tag{4.11}$$

Dove I_{max} rappresenta la somma massima per ogni istante di tempo e ϵ il fattore di decadimento dell'incentivo. In questa equazione 4.11 viene evidenziato come tali incentivi abbiano un impatto maggiore nelle fasi iniziali di creazione del pool.
Perché tale andamento fosse rispettato anche nel caso della liquidità la formula che si è ritenuta più adatta è la seguente:

$$L_{inc}(t) = L(t) \cdot (1 + \alpha(t) \cdot I(t)^{\nu}) \tag{4.12}$$

Dove:

- $\alpha(t) = \alpha \cdot e^{-\psi t}$, con α parametro fissato, ψ il tasso di decadimento
- ν che quantifica l'effetto dell'incentivo sulla liquidità

Questa relazione, come detto in precedenza, è stata testa introducendo incentivi sintetici applicati ai dati della liquidità relativi al pool WEHT/USDC. Per osservare come l'equazione 4.12 effettivamente sia in grado di modellare questo andamento con tali dati è stato riportato il grafico in figura 4.6. In quest'ultimo la linea blu rappresenta l'andamento della liquidità del pool, la linea arancione, invece, rappresenta la liquidità influenzata dagli incentivi. I parametri utilizzati sono $\alpha = 0.0001$, $\nu = 1$, $\psi = 0.002$, gli incentivi in tal caso vengono distribuiti su tutto l'arco temporale, ma possono essere ridotti ad un periodo inferiore.
Per quanto riguarda il caso degli incentivi di lockup la forma che si è ritenuta più adatta per esprimere il loro valore per ogni LP j è la seguente:

$$I_j(t) = L_j(t) \cdot \omega \cdot (t \ mod \ 30) \tag{4.13}$$

Con $L_j(t)$ liquidità del singolo LP, ω tasso di ricompensa, $t \ mod$ 30 mesi in cui è stata fornita la liquidità.
Poiché negli approcci precedenti i fornitori di liquidità sono stati considerati come un'unica entità aggregata, risulta necessario trovare un modo per combinare i contributi individuali in un'unica misura complessiva. Per questo motivo si propone di modellare gli ingressi e le uscite dal pool da parte degli LP con un processo di Poisson, seppur tali eventi non

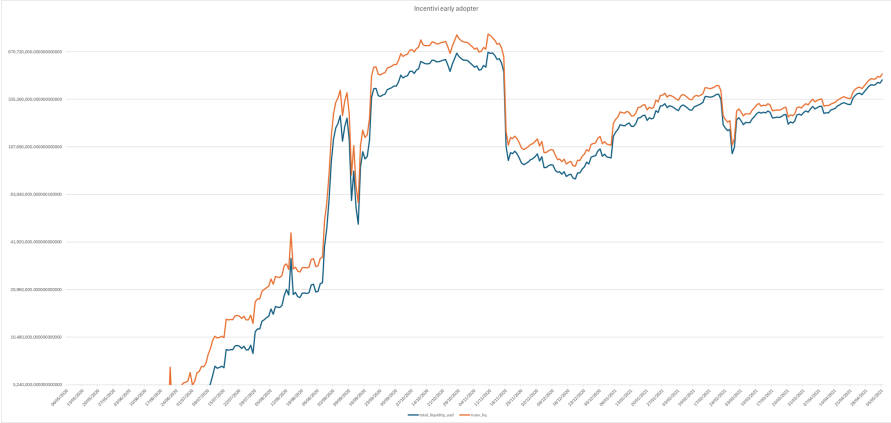


Figura 4.6. Effetto degli incentivi per early adopter

siano effettivamente indipendenti, è possibile approssimarli come tali per semplificare la trattazione. Ciò è utile perché un **processo di Poisson** è un modello stocastico utilizzato per descrivere il verificarsi di eventi in un intervallo di tempo, assumendo che questi eventi si verifichino in maniera indipendente e con una frequenza media costante. Formalmente:

Definizione: 4.2.4. *se $N(t)$ rappresenta il numero di eventi in un intervallo di lunghezza t, allora $N(t)$ segue una distribuzione di Poisson con parametro λt, dove λ è il tasso medio di eventi per unità di tempo.*

Questo modello permette, dunque, di stimare il numero atteso di LP presenti nel pool in un determinato periodo, integrando così le dinamiche di flusso all'interno del modello complessivo. Si può così ottenere una rappresentazione più accurata dell'effetto complessivo degli incentivi sulla liquidità. Per ottenere la relazione esplicita di quest'ultima con gli incentivi è importante sottolineare che conoscendo il tasso di ingresso λ e il tasso di uscita μ, allora è possibile calcolare il valore atteso di LP all'interno del pool in un dato periodo:

$$\frac{\lambda}{\mu} \cdot (1 - e^{-\mu t}) \tag{4.14}$$

Con tali informazioni è possibile calcolare anche la probabilità che un LP rimanga nel pool per un periodo superiore a un certo numero di giorni, che sarà pari a:

$$e^{-\mu t} \tag{4.15}$$

Grazie alle equazioni 4.14 e 4.15 è possibile determinare il numero medio di utenti che hanno raggiunto il lockup, ottenendo un'indicazione più precisa sul valore dell'incentivo da distribuire.
Supponendo che tale somma venga distribuita mensilmente, ci si aspetta che l'incentivo abbia un ruolo determinante in misura crescente con l'avvicinarsi della scadenza, per poi diminuire nuovamente, dando luogo a un effetto che può essere definito ciclico.

Volendo modellare questo aspetto e cercando di mantenere una forma simile a quella dell'equazione 4.12, la liquidità in funzione degli incentivi viene espressa come:

$$L_{inc}(t) = L(t) \cdot \left(1 + \alpha \cdot \left(\frac{1}{1 + e^{-k\,(d-15)}} \cdot I(t)\right)^{\nu}\right) \tag{4.16}$$

Dove α è un parametro fissato, ν misura, nuovamente, l'effetto dell'incentivo sulla liquidità e k rappresenta l'influenza del tempo sugli incentivi. Questo approccio teorico, come nel caso precedente, è supportato dall'impiego di dati sintetici, che hanno permesso di verificare la fondatezza delle ipotesi sulle relazioni tra incentivi e liquidità. Come in relazione agli incentivi per gli early adopter, anche per il lockup è stato prodotto il grafico in figura 4.7, che rende evidente la relazione espressa dall'equazione 4.16. I parametri utilizzati sono $\alpha = 0.0001$, $\nu = 1$, $\lambda = 100$, $\frac{1}{\mu} = 45$, $k = 0.3$, gli incentivi in tal caso vengono distribuiti su un arco temporale di tre mesi, ma possono essere estesi ad un periodo a piacere.

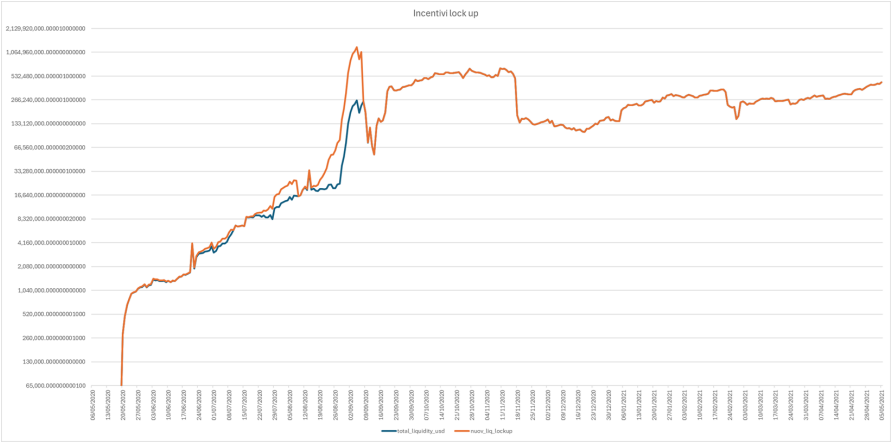


Figura 4.7. Incentivi lockup

Questo approccio ha permesso di testare diverse configurazioni ipotetiche, confermando la coerenza delle ipotesi iniziali e rafforzando la validità della forma funzionale proposta. Tale studio presenta una base teorica che potrà essere verificata in futuro con dati reali, per osservarne eventuali evidenze empiriche, che potranno ulteriormente rafforzare la comprensione complessiva del fenomeno.
Si è scelto di intraprendere questa strada poiché, seppur la liquidità possa apparire con un fattore prettamente esogeno, in realtà è fortemente influenzata dalle scelte strategiche prese in fase di sviluppo. La liquidità iniziale versata da chi apre il pool e il livello di incentivi distribuiti, infatti, possono attirare la partecipazione degli utenti. Queste scelte, infatti, possono permettere di avere maggiore controllo sulla stabilità e sulla profondità del pool. Per tale motivo si è proposto un modello che integri questi aspetti, sottolineando l'importanza delle scelte intraprese. Ciò può essere visto anche come un supporto in una fase più avanzata, quando, osservando l'effettivo andamento del pool, si possono utilizzare tali relazioni per corregerne la direzione.

Capitolo 5

Analisi dei risultati

In questo capitolo si analizzano i risultati ottenuti risolvendo il modello presentato nel paragrafo 4.2. L'obiettivo è dunque quello di comprendere quale sia la divisione ottimale del capitale tra liquidità e incentivi in fase iniziale, con particolare attenzione a questi ultimi, in modo tale che la loro distribuzione risulti utile a stimolare la partecipazione degli LP durante il periodo selezionato.
Si evidenzia come questi risultati non vogliano fornire una stima della liquidità minima necessaria affinché un liquidity pool possa mantenersi operativo, bensì indicano come il capitale destinato alla sua apertura possa essere suddiviso in modo ottimale tra liquidità e incentivi, permettendo di massimizzare l'efficacia dell'investimento iniziale. Ciò è importante poiché l'apertura di un liquidity pool è spesso inserita in progettualità di entità maggiore, che prevedono l'apporto continuo di capitali aggiuntivi e strategie di incentivazione a lungo termine. Di conseguenza, la gestione della liquidità e degli incentivi iniziali rappresenta solo una parte di un quadro più ampio volto a garantire la crescita e la sostenibilità del pool nel tempo.
L'analisi, coerentemente a quanto introdotto nella sezione 4.2.1, viene svolta in relazione agli incentivi per gli early adopter e alle ricompense successive a un periodo di lockup.
Per il primo caso viene utilizzata la funzione 4.12, suggerendo che tali incentivi avranno un effetto decrescente sulla liquidità. Al contrario nel secondo caso viene sottolineata la relazione ciclica sfruttando l'equazione 4.16. Grazie all'utilizzo di tali formule il modello dovrebbe essere in grado di evidenziare questi comportamenti, ottimizzando la distribuzione degli incentivi di conseguenza. Quello che ci si aspetta infatti, è che, nel primo caso, all'inizio del periodo di lancio del pool gli incentivi siano elevati e tendano a diminuire progressivamente. Rendendo così evidente il meccanismo studiato per premiare in maniera significativa i primi investitori, incentivando una rapida adozione del sistema.
Nel secondo caso, invece, si prevede che il modello generi una distribuzione degli incentivi con trend crescente, sottolineando come tale risultato incentivi la permanenza degli LP nel pool.
Al fine di verificare che tali ipotesi siano effettivamente supportate dai risultati, il modello è stato risolto variando i parametri. In particolare, i test svolti hanno preso in input diversi valori di capitale iniziale e di fee. Queste ultime sono state fissate a tre differenti livelli possibili: 1%, 0.3%, 0.05%. Tali valori rappresentano gli unici attualmente implementabili

sulla maggior parte delle piattaforme operative. È importante, anche, sottolineare che i dati relativi al volume sono stati tratti dai dati reali del pool WETH/USDC, garantendo così una maggiore aderenza alle dinamiche operative effettive del mercato.
In questo modo è stato possibile analizzare come scenari diversi influenzino i risultati, osservando come la distribuzione degli incentivi e la gestione della liquidità rispondano in modo specifico alle scelte adottate, individuando eventuali pattern ricorrenti.
Partendo dall'analisi dei risultati relativi alla ripartizione del capitale, emerge chiaramente come la distribuzione vari in modo significativo a seconda dei due tipi di incentivi adottati. È importante sottolineare che, in entrambi i casi, sono stati testati capitali pari a $200,000, $800,000 e $2,000,000, evidenziando schemi di distribuzione differenti e ben distinti tra early adopter e lockup.
Per i primi, i risultati puntuali sono riportati nella tabella 5.1.

Capitale	**Liquidità**	**Incentivi**
$200,000	$72,187.73	$127,812.27
$800,000	$251,657.67	$548,342.33
$2,000,000	$629,144.18	$1,370,855.82

Tabella 5.1. Divisione ottimale con incentivi per early adopter

Il modello, in questo contesto, ha ottimizzato la ripartizione del capitale favorendo gli incentivi, a discapito di una liquidità inferiore. Dai dati riportati in tabella, infatti, si può osservare come in tutti e tre i casi si ha che le proporzioni rimangono siano simili, indipendentemente dal valore totale del capitale. Ciò suggerisce che una distribuzione ottimale si ha nel caso in cui circa 35% del capitale è destinato alla liquidità e il restante 65% agli incentivi, rendendo questi ultimi significativamente più elevati.
Relativamente al caso degli incentivi di lockup la situazione risulta essere nettamente differente. I risultati di questa ottimizzazione sono stati raccolti all'interno della tabella 5.2.

Capitale	**Liquidità**	**Incentivi**
$200,000	$113,151.96	$86,848.04
$800,000	$763,617.63	$36,382.37
$2,000,000	$1,909,044.07	$90,955.93

Tabella 5.2. Divisione ottimale con incentivi di lockup

In questo caso i risultati del modello suggeriscono un approccio opposto rispetto a quello

precedente, ossia favorendo la liquidità. Quest'ultima, come si può osservare dalla tabella dei risultati, risulta che in proporzione nettamente maggiore, raggiungendo il 95.5% nel secondo e terzo caso, contro il 4.5% rappresentato dagli incentivi. Nel primo caso, invece, la distinzione appare meno evidente, suggerendo che, in condizioni di liquidità non elevata, sia necessario un livello più alto di incentivi.
Da tale analisi, dunque, emerge un pattern molto definito nella ripartizione del capitale, differenziato in base al tipo di incentivo adottato. Viene favorita una quota significativa di incentivi per stimolare l'ingresso in fase iniziale nel caso relativo agli early adopter, al contrario viene privilegiata la liquidità, nel caso del lockup, che porta a maggiore stabilità.
A seguito di questa analisi si può fornire la seguente interpretazione dei risultati: nel primo caso, il modello suggerisce di offrire incentivi molto alti per attirare gli LP, facilitando la creazione di una base solida. In questo modo, l'incremento della liquidità diventa una conseguenza naturale della partecipazione di un numero consistente di LP, consentendo di investire inizialmente una quantità inferiore di capitale e di raggiungere progressivamente una maggiore stabilità operativa.
Nel secondo caso, invece, l'evidente predominanza della liquidità rispetto agli incentivi può essere interpretata come la ricerca di una maggiore stabilità iniziale, in quanto il modello non mira ad attrarre nuovi LP tramite incentivi elevati, bensì l'obiettivo è quello di consolidare il capitale già presente nel pool. Gli incentivi, di conseguenza, sono impiegati esclusivamente per disincentivare l'uscita degli investitori, contribuendo così a mantenere la stabilità del sistema nel tempo, risultando necessari in misura maggiore quando la liquidità è più limitata.
In sintesi, i due risultati evidenziano strategie di distribuzione del capitale radicalmente diverse, mentre nel primo scenario l'obiettivo è attirare e consolidare rapidamente la partecipazione degli LP tramite incentivi elevati, nel secondo scenario si privilegia la stabilità a lungo termine, favorendo la permanenza dei partecipanti attraverso una maggiore allocazione di liquidità e un minore ricorso agli incentivi. Questi pattern distinti, evidenziati dai dati, confermano che la strategia di distribuzione degli incentivi segue uno schema funzionale preciso, rispondendo in modo coerente alle differenti esigenze di incentivazione.
In condizioni di liquidità bassa, tuttavia, possono emergere analogie tra i due approcci, poiché la necessità di incentivare la partecipazione diventa più rilevante. Questo offre uno spunto per una possibile estensione, approfondita nella sezione 5.1, dove viene introdotta la possibilità di considerare entrambi gli incentivi durante l'ottimizzazione. Ciò risulta particolarmente interessante dopo aver tracciato le curve di comportamento relative al caso degli early adopter e del lockup, comprendendo come vari la distribuzione tra liquidità e incentivi per diversi livelli di capitale iniziale.
Un ulteriore obiettivo di tale modello, come sottolineato in precedenza, consiste nel fornire indicazioni precise sulla ripartizione temporale della somma destinata agli incentivi. L'ottimizzazione di quest'ultimi mira sempre massimizzare la funzione di utilità definita nel paragrafo 4.2, permettendo di verificare che anche tali risultati siano coerenti con quelli attesi. Il modello fornisce, dunque, una chiave interpretativa fondamentale, rivelando come la distribuzione degli incentivi non avvenga in modo casuale, ma segua uno schema preciso che si adegua alle variazioni del volume, ottimizzando così la partecipazione degli LP e la gestione complessiva del capitale investito.
L'analisi continua a seguire la distinzione di incentivi utilizzata in precedenza.

I risultati ottenuti, per il caso degli early adopter, sono rappresentati graficamente nella figura 5.1. Tale grafico riporta il caso in cui il capitale è fissato a $800,000, ma negli altri casi i risultati sono analoghi.

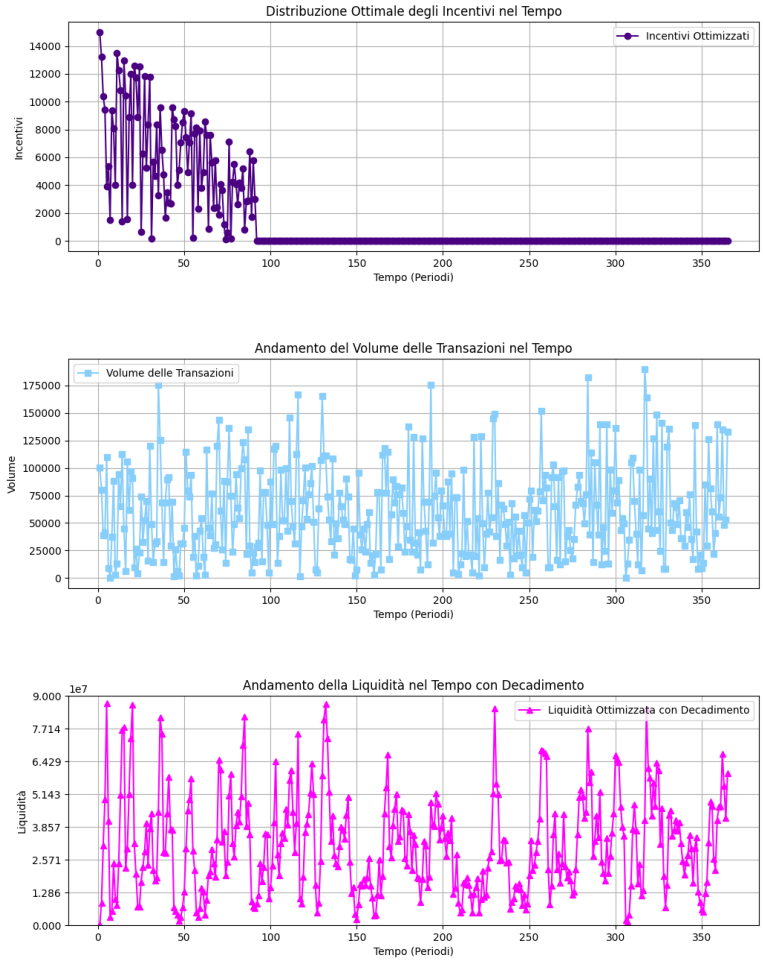


Figura 5.1. Risultati con incentivi per early adopter

Analizzando tali dati risulta evidente che l'andamento degli incentivi sia coerente con le aspettative, seguendo un andamento decrescente. Si osserva, inoltre, che la distribuzione degli incentivi è stata prevista esclusivamente per i primi tre mesi, evidenziando una strategia che concentra la spinta incentivante nelle fasi iniziali per poi attenuarsi progressivamente.
Un aspetto particolarmente rilevante, che emerge analizzando i risultati forniti dal modello, è il fatto gli incentivi sono correlati negativamente con il volume (figura 5.2, in riferimento a capitale pari a $800,000). Allo stesso tempo è possibile osservare come, invece, quest'ultimo sia correlato positivamente con la liquidità. Tali informazioni risultano

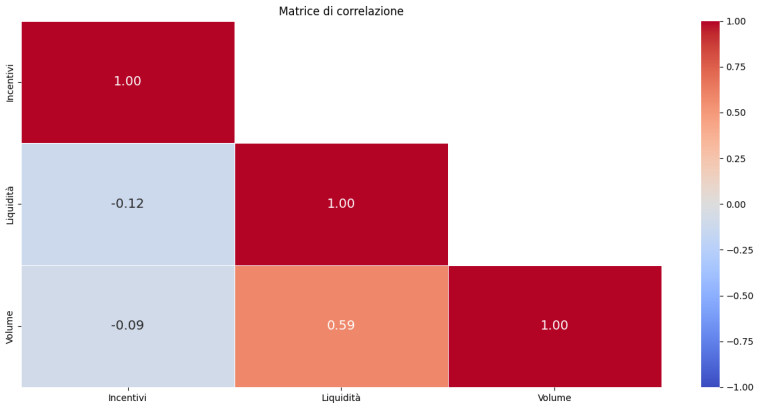


Figura 5.2. Matrice di correlazione con incentivi per early adopter

fondamentali per l'interpretazione dei risultati. Ciò suggerisce che all'aumentare dell'attività di trading, la quantità di incentivi erogata tende a diminuire, mentre in periodi caratterizzati da un volume inferiore, l'incremento degli incentivi diventa cruciale per stimolare l'adesione e mantenere attivo il pool.
Concentrandosi, invece, sulla correlazione tra liquidità e volume si osserva che al crescere del volume, la liquidità aumenta, confermando che un elevato livello di attività contribuisce direttamente a rafforzare la stabilità del pool, riducendo così la necessità di introdurre nuovi incentivi per attrarre LP. Queste dinamiche permettono di comprendere come il modello sia in grado di registrare eventuali variazioni di mercato, suggerendo una soluzione che si possa adattare al meglio alla situazione.
Risulta, dunque, che la distribuzione degli incentivi non è casuale, ma segue logiche ben precise, legate all'andamento del volume di trading, permettendo di garantire una crescita stabile del pool.
Per visualizzare queste relazioni anche in modo grafico, in appendice F è stata inserita la figura F.1 che rappresenta l'andamento delle variabili chiave del modello, generata utilizzando un numero inferiore di periodi di ottimizzazione, mantenendo comunque la struttura del modello invariata. Questo permette di evidenziare con maggiore immediatezza le dinamiche tra incentivi, volume e liquidità, offrendo così un ulteriore supporto alla comprensione delle correlazioni emerse dall'analisi.
In relazione agli incentivi di lockup si possono fare osservazioni analoghe, analizzando i dati riportati nel grafico in figura 5.3, anche in questo caso relativi al capitale iniziale di $800,000.
Tali risultati risultano nuovamente coerenti con la relazione che ci sia aspettava per questo tipo di incentivi. La distribuzione, infatti, avviene mensilmente e si può osservare un andamento crescente a livello di valore. Anche in questo caso è interessante come, facendo riferimento alla matrice di correlazione in figura 5.4 (capitale iniziale $800,000, sia possibile evidenziare le stesse relazioni seppur meno significative. Ciò può essere attribuito al fatto che gli incentivi, essendo presenti in quantità minore, hanno un impatto sul sistema meno rilevante. Nonostante questo, si può comunque assumere che il modello agisca in

Figura 5.3. Risultati con incentivi di lockup

modo analogo al caso precedente, adattandosi sempre all'andamento di mercato. Anche in questo caso in appendice F è stata riportata la figura F.1 per poter osservare meglio queste relazioni graficamente.
Per ottenere tali risultati è stato utilizzato il codice in appendice E, in cui, a tale modello, è stata aggiunta una funzione di rinforzo per evidenziare maggiormente le relazioni tra liquidità e volume. Questa funzione permette di ottenere un volume non puramente casuale, ma che sia in grado di riflettere l'effetto positivo dato dalla liquidità. Ciò risulta in un incremento maggiore del volume quando la liquidità è elevata, viceversa sarà minore in presenza di livelli inferiori di liquidità, modellando in modo realistico l'interazione tra queste due variabili.
Per quanto riguarda l'IL invece, il modello, servendosi della formula 3.15, ne calcola una media sull'intervallo temporale compreso tra 0 e t. Le uscite, inoltre, vengono approssimate da un processo di Poisson, come spiegato nel paragrafo 4.2.1. Combinando questi

due aspetti è possibile garantire una rappresentazione realistica dell'effettivo valore della perdita impermanente.
Come introdotto all'inizio di questo capitolo, l'analisi è stata svolta anche considerando differenti livelli di fee. Il cambiamento di tale parametro, tuttavia, non ha apportato differenze significative nei risultati, che sono rimasti sostanzialmente invariati. Questo fenomeno è implicabile al fatto che l'ordine di grandezza delle fee risulta trascurabile rispetto a quello di liquidità e volume, suggerendo che la strategia ottimale da adottare risulta robusta e non subisce modifiche sostanziali al variare di tali livelli.
In sintesi, l'analisi dei risultati rivela un quadro completo, in cui la scelta tra incentivi per early adopter e lockup determina dinamiche differenti nella distribuzione del capitale. Rappresentando un passo iniziale nella comprensione delle dinamiche di gestione dei liquidity pool e fornendo le basi per sviluppi futuri.
È importante sottolineare che i risultati ottenuti durante questa analisi possono essere influenzati in modo significativo dai parametri di inizializzazione e dall'algoritmo di ottimizzazione scelto. La configurazione iniziale e la metodologia adottata per la ricerca della soluzione possono incidere sulla convergenza e sulla qualità della soluzione finale, introducendo variabilità nei risultati. Pertanto, tali aspetti devono essere considerati come elementi chiave nella valutazione della robustezza del modello e nell'interpretazione dei risultati ottenuti.

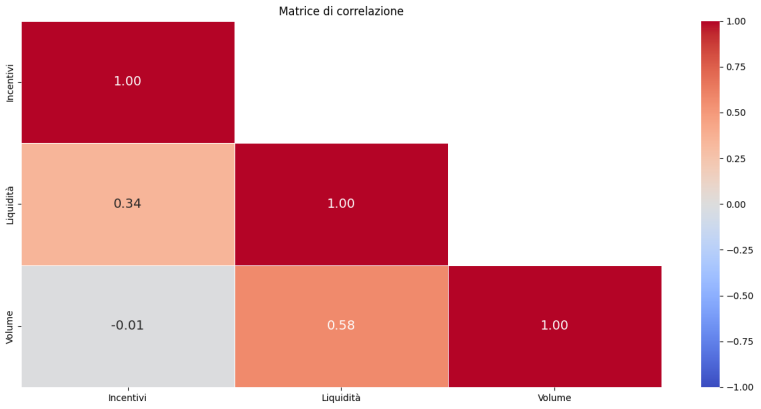


Figura 5.4. Matrice di correlazione con incentivi di lockup

5.1 Estensioni future

Il modello introdotto ha offerto risultati interessanti ai fini dello studio, consentendo di comprendere come la distribuzione della liquidità e degli incentivi possa influenzare la stabilità del pool. Esistono, però, ulteriori approfondimenti che possono essere implementati, per ottenere soluzioni sempre più accurate in grado di modellare ancora meglio le reali dinamiche di mercato.
Una possibile estensione iniziale potrebbe consistere nell'integrare il livello di fee all'interno

del processo di ottimizzazione del modello, implementando una procedura che determini il livello di fee ottimale in grado di massimizzare la funzione di utilità. Ciò permette di evidenziare eventuali interazioni dinamiche con la liquidità, il volume e gli incentivi.
In relazione a questi ultimi, è possibile ampliare le prospettive del modello per renderlo più realistico, introducendo diverse tipologie di incentivi che possono attivarsi in momenti differenti dell'ottimizzazione. Questa estensione permetterebbe di analizzare in maniera più dettagliata come una combinazione di incentivi distribuiti in fasi differenti possa influenzare l'adozione iniziale, la stabilità del pool e l'accesso degli LP. Si potrebbero, ad esempio, prevedere incentivi più consistenti nelle fasi iniziali per attrarre i primi investitori, per poi sostituirli con premi di lockup, per favorire la permanenza degli LP.
Nell'implementazione attuale, inoltre, la generazione dei prezzi è basata sui dati reali ricavati dall'analisi del pool WETH/USDC, ma risulta possibile introdurre modelli stocastici più complessi, per garantire un'approssimazione migliore. Tra questi il moto browniano e i processi di Lévy possono essere ottimi candidati. Il primo risulta fondamentale per modellare la volatilità, i secondi per catturare anche gli eventi estremi, molto comuni in questo contesto.
Ulteriori analisi, che potrebbero fornire risultati interessanti, sono relative all'implementazione di AMM differenti, come ad esempio quello a liquidità concentrata. In questo modo risulterebbe possibile valutare come la concentrazione del capitale influisca sui parametri finora studiati.
È possibile, inoltre, applicare algoritmi di statistica bayesiana per una ricerca più accurata della forma funzionale della liquidità, con introduzione di parametri che si adattino meglio a situazioni differenti.
Implementando queste estensioni è possibile avere un quadro più completo relativo al setup, tenendo conto della gestione del capitale in un contesto di mercato più complesso.

Capitolo 6

Conclusione

Negli ultimi anni, la DeFi ha rivoluzionato il panorama finanziario, offrendo un'alternativa innovativa ai tradizionali sistemi bancari e di intermediazione. Grazie all'uso della tecnologia blockchain, la DeFi consente la creazione di ecosistemi finanziari autonomi, trasparenti e accessibili a livello globale. Tra gli strumenti chiave di questa rivoluzione emergono i liquidity pool, fondamentali per garantire liquidità ai mercati decentralizzati e facilitare il funzionamento degli AMM.
Questo elaborato, dunque, ha fornito un'analisi approfondita della DeFi applicata nel contesto della blockchain ponendo particolare attenzione ai liquidity pool, con l'obiettivo di garantire una guida pratica da poter seguire in fase di setup.
Per raggiungere questo scopo è stato dettagliatamente introdotto il contesto in cui è stata svolta l'analisi. Nel capitolo 2, infatti, è stato approfondito il concetto di blockchain, soffermandosi su come la sua evoluzione abbia portato a favorire lo sviluppo della DeFi. Sono stati inoltre analizzati i token e gli smart contract, al fine di sottolineare il loro ruolo centrale nella DeFi. Ciò ha permesso di evidenziare i principali vantaggi e le potenzialità di un contesto finanziario privo di intermediari rispetto alla finanza tradizionale.
Nel terzo capitolo (3) i liquidity pool sono stati posti al centro dell'analisi, evidenziando la necessità di implementare un meccanismo decentralizzato in grado di sostituire gli order book. Di conseguenza sono state analizzate le diverse tipologie di AMM, come quelli a prodotto costante, a somma costante, a media costante, gli AMM ibridi e gli AMM a liquidità concentrata. Oltre agli aspetti positivi, però, sono stati evidenziati anche quelli negativi, come l'IL, che rappresenta una delle maggiori criticità di questo strumento della DeFi.
Si è arrivati così a definire chiaramente lo scopo di tale ricerca nel capitolo 4. Ciò ha permesso di introdurre la metodologia seguita nella definizione del processo che ha portato a raggiungere i risultati voluti. Per questo motivo, infatti, è stato definito accuratamente il flusso logico necessario agli sviluppatori nei momenti cruciali di definizione delle caratteristiche del pool. Tale processo è stato integrato con una trattazione, dal punto di vista matematico, di un modello che potesse rappresentare in modo efficiente le relazioni che intercorrono tra le diverse variabili, quali volume e liquidità. Per quest'ultima è stata prevista anche la ricerca di una forma funzionale che ne rappresentasse le caratteristiche, servendosi di dati reali integrati ragionamenti puramente teorici.

Grazie a questi elementi è stato possibile risolvere il modello e analizzarne i risultati nel capitolo 5. Questo ha permesso di evidenziare alcuni pattern, giungendo alla conclusione che il tipo di incentivi influenzi fortemente il livello di liquidità necessario in fase di apertura. Si è osservato anche, come il modello sia in grado di reagire ad eventi esterni regolando di conseguenza la distribuzione degli incentivi.
Questi risultati hanno permesso di rispondere alle domande di ricerca che sono state poste. L'orizzonte però può comunque essere ampliato con diverse estensioni del modello, portando ad avere risultati più realistici e robusti.
Questa tesi, dunque, ha fornito un quadro analitico approfondito che evidenzia l'importanza di una gestione dinamica e ottimale degli incentivi nei liquidity pool, offrendo un contributo significativo per migliorare l'efficienza e la stabilità dei mercati decentralizzati. Le analisi condotte e i risultati ottenuti offrono una solida base per ulteriori sviluppi e approfondimenti, aprendo la strada a nuove soluzioni per la gestione del capitale e degli incentivi in contesti di mercato in continua evoluzione.

Bibliografia

Arman Abgaryan, Utkarsh Sharma, and Joshua Tobkin. Proof of efficient liquidity: A staking mechanism for capital efficient liquidity. *arXiv preprint arXiv:2401.04521*, 2024.

Ittai Abraham, Srinivas Devadas, Danny Dolev, Kartik Nayak, and Ling Ren. Efficient synchronous byzantine consensus. *arXiv preprint arXiv:1704.02397*, 2017.

Rebecca Abraham. A formulation of investor sentiment of cryptocurrencies and cryptocurrency futures and options. *Theoretical Economics Letters*, 14(2):597–616, 2024.

Arnold Mashud Abukari, Vivek Gupta, Jhansi Bharathi Madavarapu, and Vijaya Kittu Manda. A homomorphic block approach to blockchain and cloud erp implementation. *Journal of Applied Intelligent Systems and Information Sciences*, 4(1):50–59, 2023.

Austin Adams, Benjamin Y Chan, Sarit Markovich, and Xin Wan. Don't let mev slip: the costs of swapping on the uniswap protocol. In *International Conference on Financial Cryptography and Data Security*, pages 172–191. Springer, 2024.

Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. *Tech. rep., Uniswap, Tech. Rep.*, 2021.

Guillermo Angeris, Tarun Chitra, and Alex Evans. When does the tail wag the dog? curvature and market making. 2022.

Andreas M Antonopoulos and David A Harding. *Mastering bitcoin*. " O'Reilly Media, Inc.", 2023.

Adam Back et al. Hashcash-a denial of service counter-measure. 2002.

Massimo Bartoletti, James Hsin-yu Chiang, and Alberto Lluch Lafuente. Sok: lending pools in decentralized finance. In *Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin, DeFi, VOTING, and WTSC, Virtual Event, March 5, 2021, Revised Selected Papers 25*, pages 553–578. Springer, 2021.

Abdeljalil Beniiche. A study of blockchain oracles. *arXiv preprint arXiv:2004.07140*, 2020.

Maxim Bichuch and Zachary Feinstein. Axioms for automated market makers: A mathematical framework in fintech and decentralized finance. *arXiv preprint arXiv:2210.01227*, 2022.

Tomas Björk. *Arbitrage theory in continuous time*. Oxford university press, 2009.

Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, et al. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. *Chainlink Labs*, 1:1–136, 2021.

Vitalik Buterin et al. Ethereum white paper. *GitHub repository*, 1:22–23, 2013.

Agostino Capponi, Ruizhe Jia, and Brian Zhu. The paradox of just-in-time liquidity in decentralized exchanges: More providers can lead to less liquidity. *Available at SSRN 4648055*, 2024.

Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OsDI*, volume 99, pages 173–186, 1999.

Colin Chan. dydx: Liquidity providers' incentive programme review. *arXiv preprint arXiv:2307.03935*, 2023.

Erdong Chen, Mengzhong Ma, and Zixin Nie. Perpetual future contracts in centralized and decentralized exchanges: Mechanism and traders' behavior. *Electronic Markets*, 34 (1):35, 2024.

Wayne Chen, Songwei Chen, and Preston Rozwood. Improving capital efficiency and impermanent loss: Multi-token proactive market maker. *arXiv preprint arXiv:2309.00632*, 2023.

Joseph Clark. The replicating portfolio of a constant product market. *Available at SSRN 3550601*, 2020.

Wei Dai. B-money, 1998.

Richard Dewey and Craig Newbold. The pricing and hedging of constant function market makers. *arXiv preprint arXiv:2306.11580*, 2023.

Fayçal Drissi. Models of market liquidity: Applications to traditional markets and automated market makers. *Available at SSRN 4424010*, 2023.

Michael Egorov. Stableswap-efficient mechanism for stablecoin liquidity. *Retrieved Feb*, 24:2021, 2019.

Srisht Fateh Singh, Vladyslav Nekriach, Panagiotis Michalopoulos, Andreas Veneris, and Jeffrey Klinck. Option contracts in the defi ecosystem: Opportunities, solutions, and technical challenges. *International Journal of Network Management*, 35(2):e70005, 2025.

Hal Finney. Reusable proofs of work. *Web Archives Homepage*, 2004.

Pierluigi Freni, Enrico Ferro, and Roberto Moncada. Tokenomics and blockchain tokens: A design-oriented morphological framework. *Blockchain: Research and Applications*, 3 (1):100069, 2022. ISSN 2096-7209.

Robin Fritsch. Concentrated liquidity in automated market makers. In *Proceedings of the 2021 ACM CCS Workshop on Decentralized Finance and Security*, pages 15–20, 2021.

Masaaki Fukasawa, Basile Maire, and Marcus Wunsch. Model-free hedging of impermanent loss in geometric mean market makers. *Available at SSRN 4397904*, 2023.

Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 3–16, 2016.

Matthias Hafner and Helmut Dietl. Impermanent loss conditions: An analysis of decentralized exchange platforms. *arXiv preprint arXiv:2401.07689*, 2024.

Joel Hasbrouck, Thomas J Rivera, and Fahad Saleh. An economic model of a decentralized exchange with concentrated liquidity. *Available at SSRN 4529513*, 2023.

Xue Dong He, Chen Yang, and Yutian Zhou. Liquidity pool design on automated market makers. *arXiv preprint arXiv:2404.13291*, 2024a.

Xue Dong He, Chen Yang, and Yutian Zhou. Optimal design of automated market makers on decentralized exchanges. *arXiv preprint arXiv:2404.13291*, 2024b.

Lioba Heimbach, Eric Schertenleib, and Roger Wattenhofer. Risks and returns of uniswap v3 liquidity providers. In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, pages 89–101, 2022.

Eric Hughes. A cypherpunk's manifesto. In *The electronic privacy papers: Documents on the battle for privacy in the age of surveillance*, pages 285–287. 1997.

Johannes Rude Jensen, Mohsen Pourpouneh, Kurt Nielsen, and Omri Ross. The homogenous properties of automated market makers. *arXiv preprint arXiv:2105.02782*, 2021.

Ariston Karagiorgis, Antonis Ballis, and Konstantinos Drakos. The skewness-kurtosis plane for cryptocurrencies' universe. *International Journal of Finance & Economics*, 29(2):2543–2555, 2024.

Adam Khakhar and Xi Chen. Delta hedging liquidity positions on automated market makers. *arXiv preprint arXiv:2208.03318*, 2022.

Hyoung Joong Kim, Soohyuk Choi, Yong Tae Yoon, and Shiyong Yoo. Impermanent loss and gain of automated market maker smart contracts. *Authorea Preprints*, 2022.

Hyoung Joong Kim, Gyu M Lee, Jongwon Lee, Sora Kang, Seong Wook Chae, and Jun-Seok Park. A comparison of impermant loss for various cfmms. In *2024 IEEE International Conference on Blockchain (Blockchain)*, pages 542–548. IEEE, 2024.

Bhaskar Krishnamachari, Qi Feng, and Eugenio Grippo. Dynamic curves for decentralized autonomous cryptocurrency exchanges. *arXiv preprint arXiv:2101.02778*, 2021.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the works of leslie lamport*, pages 203–226. 2019.

Alfred Lehar and Christine A Parlour. Decentralized exchange: The uniswap automated market maker. *Available at SSRN 3905316*, 2021.

Thomas LI, Siddharth Naik, Andrew Papanicolaou, and Lorenzo Schoenleber. Implied impermanent loss: A cross-sectional analysis of decentralized liquidity pools. *Available at SSRN 4811111*, 2024.

Alexander Lipton, Vladimir Lucic, and Artur Sepp. Unified approach for hedging impermanent loss of liquidity provision. *arXiv preprint arXiv:2407.05146*, 2024.

Damiano Di Francesco Maesa and Paolo Mori. Blockchain 3.0 applications survey. *Journal of Parallel and Distributed Computing*, 138:99–114, 2020.

Fernando Martinelli and Nikolai Mushegian. A non-custodial portfolio manager, liquidity provider, and price sensor. *URl: https://balancer. finance/whitepaper*, 2019.

Jovanka Lili Matic, Natalie Packham, and Wolfgang Karl Härdle. Hedging cryptocurrency options. *Review of Derivatives Research*, 26(1):91–133, 2023.

Timothy C May. The crypto anarchist manifesto. 1988.

Jason Milionis, Ciamac C Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing. *arXiv preprint arXiv:2208.06046*, 2022.

Vijay Mohan. Automated market makers and decentralized exchanges: a defi primer. *Financial Innovation*, 8(1):20, 2022.

Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Satoshi Nakamoto*, 2008. URL http://www.bitcoin.org/bitcoin.pdf.

Harish Natarajan, Solvej Krause, and Helen Gradstein. *Distributed ledger technology and blockchain*. World Bank, 2017.

Abraham Othman and Tuomas Sandholm. Automated market makers that enable new settings: Extending constant-utility cost functions. In *International Conference on Auctions, Market Mechanisms and Their Applications*, pages 19–30. Springer, 2011.

Andrea Pinna and Wiebe Ruttenberg. Distributed ledger technologies in securities post-trading revolution or evolution? *ECB Occasional Paper*, (172), 2016.

Alexander Port and Neelesh Tiruviluamala. Mixing constant sum and constant product market makers. *arXiv preprint arXiv:2203.12123*, 2022.

Mohsen Pourpouneh, Kurt Nielsen, and Omri Ross. Automated market makers. Technical report, IFRO Working Paper, 2020.

Dan Robinson. Uniswap v3: The Universal AMM — paradigm.xyz. https://www.paradigm.xyz/2021/06/uniswap-v3-the-universal-amm, 2021. [Accessed 17-02-2025].

Fabian Schär. Decentralized finance: On blockchain-and smart contract-based financial markets. *FRB of St. Louis Review*, 2021.

Don DH Shin. Blockchain: The emerging technology of digital trust. *Telematics and informatics*, 45:101278, 2019.

Melanie Swan. *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

Nick Szabo. Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought,(16)*, 18(2):28, 1996.

Nick Szabo. Bit gold. *Recuperado de https://nakamotoinstitute. org/bit-gold/TVer página*, pages 251–260, 2005.

Mostafa Tamandi. Modeling bitcoin price dynamics: Overcoming kurtosis and skewness challenges for enhanced predictive accuracy. *Computational Economics*, pages 1–16, 2024.

Rohan Tangri, Peter Yatsyshin, Elisabeth A Duijnstee, and Danilo Mandic. Generalizing impermanent loss on decentralized exchanges with constant function market makers. *arXiv preprint arXiv:2301.06831*, 2023.

Don Tapscott and R Kirkland. How blockchains could change the world. *Retrieved on March*, 15(3):2017, 2016.

Don Tapscott and Alex Tapscott. Blockchain revolution: How the technology behind bitcoin and other cryptocurrencies is changing the world. 2016. URL https://api.semanticscholar.org/CorpusID:201446520.

Shermin Voshmgir. *Token economy: How the Web3 reinvents the internet*, volume 2. Token Kitchen, 2020.

Yongge Wang. Automated market makers for decentralized finance (defi). *arXiv preprint arXiv:2009.01676*, 2020.

Brian Z Zhu, Dingyue Liu, Xin Wan, Gordon Liao, Ciamac C Moallemi, and Brad Bachu. What drives liquidity on decentralized exchanges? evidence from the uniswap protocol. *arXiv preprint arXiv:2410.19107*, 2024.

Damian Zięba. Lévy processes in the cryptocurrency market. *Available at SSRN 4900228*, 2019.

Appendice A

Codice per l'estrazione della liquidità

```
import time
import csv
import requests
from web3 import Web3
from datetime import datetime, timedelta

print(requests)

# Connettersi alla rete principale di Ethereum tramite Infura
INFURA_URL = "https://mainnet.infura.io/v3/insert_API"
w3 = Web3(Web3.HTTPProvider(INFURA_URL))

# Indirizzo del contratto e ABI
contract_address = "insert_address"
contract_abi = [
    {
        "constant": True,
        "inputs": [],
        "name": "getReserves",
        "outputs": [
            {"name": "_reserve0", "type": "uint112"},
            {"name": "_reserve1", "type": "uint112"},
            {"name": "_blockTimestampLast", "type": "uint32"}
        ],
        "payable": False,
        "stateMutability": "view",
        "type": "function"
    }
]

# Inizializzare il contratto
```

```
contract = w3.eth.contract(address=contract_address, abi=contract_abi)

# Chiave API di CryptoCompare
CRYPTOCOMPARE_API_KEY = "your_API"  # Sostituire con la propria chiave API di
↪ CryptoCompare

# Funzione per ottenere il numero di blocco da una data specifica
def get_block_from_date(target_date, max_retries=8, delay=1):
    etherscan_api_url = "https://api.etherscan.io/api"
    params = {
        "module": "block",
        "action": "getblocknobytime",
        "timestamp": int(target_date.timestamp()),  # Convertire in timestamp
        ↪ UNIX
        "closest": "before",
        "apikey": "your_API"
    }

    for attempt in range(max_retries):
        try:
            response = requests.get(etherscan_api_url, params=params)
            response.raise_for_status()
            data = response.json()
            if data["status"] == "1":  # Risposta di successo
                return int(data["result"])
            else:
                raise ValueError(f"Errore API: {data['message']}")
        except requests.exceptions.RequestException as e:
            print(f"Tentativo {attempt + 1} fallito: {e}. Riprovo tra {delay}
            ↪ secondi...")
            time.sleep(delay)
            delay *= 2  # Ritardo esponenziale
    raise ValueError(f"Impossibile recuperare il blocco per {target_date} dopo
    ↪ {max_retries} tentativi.")

# Funzione per ottenere il prezzo ETH/USD in una data specifica utilizzando
↪ CryptoCompare
def get_historical_eth_price(date, max_retries=8, delay=1):
    """
    Recupera il prezzo storico ETH/USD per una data specifica utilizzando
    ↪ CryptoCompare.

    Args:
        date (str): La data nel formato 'gg-mm-aaaa'.

    Returns:
        float: Il prezzo ETH/USD nella data fornita.
    """
    url = f"https://min-api.cryptocompare.com/data/v2/histoday"
```

```
    params = {
        "fsym": "ETH",  # Simbolo Ethereum
        "tsym": "USD",  # Simbolo Dollaro USA
        "toTs": int(datetime.strptime(date, "%d-%m-%Y").timestamp()),  #
        ↪  Convertire la data in timestamp
        "limit": 1,  # Abbiamo bisogno solo di un punto dati
        "api_key": CRYPTOCOMPARE_API_KEY  # La tua chiave API di CryptoCompare
    }

    for attempt in range(max_retries):
        try:
            response = requests.get(url, params=params)
            response.raise_for_status()
            data = response.json()
            if data["Response"] == "Success" and "Data" in data["Data"]:
                price_data = data["Data"]["Data"][0]  # Ottenere la prima (e
                ↪  unica) voce
                return price_data["close"]  # Prezzo di chiusura per ETH/USD
            else:
                raise ValueError("Errore nel recupero del prezzo ETH da
                ↪  CryptoCompare")
        except requests.exceptions.RequestException as e:
            print(f"Tentativo {attempt + 1} fallito: {e}. Riprovo tra {delay}
            ↪  secondi...")
            time.sleep(delay)
            delay *= 2  # Ritardo esponenziale
    raise ValueError(f"Impossibile recuperare il prezzo ETH per {date} dopo
    ↪  {max_retries} tentativi.")

# Definire l'intervallo di date
start_date = datetime(2020, 5, 6)
end_date = datetime(2025, 1, 26)

# Aprire il file CSV per la scrittura
with open("liquidity_data_with_eth_price_4anni.csv", "w", newline="") as
↪  csvfile:
    fieldnames = ["date", "block", "usdc_reserve", "eth_reserve",
    ↪  "eth_usd_price", "total_liquidity_usd"]
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()

    # Ciclo per ogni giorno e recupero dei dati di liquidità
    current_date = start_date
    while current_date <= end_date:
        try:
            # Ottenere il numero di blocco per la data corrente
            block_number = get_block_from_date(current_date)

            # Recuperare le riserve al blocco
```

```
            reserves = contract.functions.getReserves().call(block_identifier
            =block_number)
            reserve0 = reserves[0]  # Riserva USDC (6 decimali)
            reserve1 = reserves[1]  # Riserva ETH (18 decimali)

            # Ottenere il prezzo ETH/USD per la data corrente
            eth_usd_price =
            ↪  get_historical_eth_price(current_date.strftime("%d-%m-%Y"))

            # Calcolare la liquidità totale in USD
            usdc_value = reserve0 / 1e6  # Convertire in unità standard (USDC)
            eth_value_in_usd = (reserve1 / 1e18) * eth_usd_price  # Convertire
            ↪  ETH in USD
            total_liquidity_usd = usdc_value + eth_value_in_usd

            # Scrivere i risultati nel CSV
            writer.writerow({
                "date": current_date.strftime("%Y-%m-%d"),
                "block": block_number,
                "usdc_reserve": usdc_value,
                "eth_reserve": reserve1 / 1e18,
                "eth_usd_price": eth_usd_price,
                "total_liquidity_usd": total_liquidity_usd
            })

            print(f"{current_date.strftime('%Y-%m-%d')} - Blocco {block_number}
            ↪  - USDC: {usdc_value} - ETH: {reserve1 / 1e18} - ETH/USD:
            ↪  {eth_usd_price} - Liquidità Totale: {total_liquidity_usd}")

        except Exception as e:
            print(f"Errore su {current_date.strftime('%Y-%m-%d')}: {e}")

        # Incrementare la data di un giorno
        current_date += timedelta(days=1)
```

Appendice B

Codice per l'estrazione del volume

```
import csv
from dune_client.client import DuneClient

# Inizializza il client di Dune
dune = DuneClient("your_API")  # Sostituisci con la tua chiave API

# ID della query
query_id = query_number  # Sostituisci con il tuo ID query

# Ottieni i risultati della query
query_result = dune.get_latest_result(query_id)

# Verifica se i risultati sono validi
if query_result.result:
    # Nome del file CSV in cui salvare i risultati
    output_file = "query_results1.csv"

    # Estrarre le righe dei dati
    rows = query_result.result.rows
    if rows:
        # Ottieni i nomi delle colonne
        headers = rows[0].keys()

        # Scrivi i risultati nel file CSV
        with open(output_file, mode="w", newline="", encoding="utf-8") as file:
            writer = csv.DictWriter(file, fieldnames=headers)
            writer.writeheader()
            writer.writerows(rows)

        print(f"Risultati salvati con successo in '{output_file}'.")
    else:
```

```
        print("Nessun risultato trovato nella query.")
else:
    print("Errore nell'ottenere i risultati della query.")
```

La query è definita nel seguente modo su Dune Analytics:

```
WITH
filtered_trades AS (
    SELECT
        block_date,
        CASE
            WHEN token_bought_symbol = 'USDC' THEN token_bought_amount
            WHEN token_sold_symbol = 'USDC' THEN token_sold_amount
            ELSE 0
        END AS volume_usdc,
        CASE
            WHEN token_bought_symbol = 'ETH' THEN token_bought_amount
            WHEN token_sold_symbol = 'ETH' THEN token_sold_amount
            ELSE 0
        END AS volume_eth,
        1 AS swap_count
    FROM
        uniswap_v2_ethereum.trades
    WHERE
        project_contract_address = CAST(
            FROM_HEX('B4e16d0168e52d35CaCD2c6185b44281Ec28C9Dc') AS varbinary
        )
        AND block_date BETWEEN DATE '2020-05-06' AND DATE  '2025-01-26'
)
SELECT
    block_date,
    SUM(volume_usdc) AS trading_volume_usdc,
    SUM(volume_eth) AS trading_volume_eth,
    SUM(swap_count) AS number_of_swaps
FROM
    filtered_trades
GROUP BY
    block_date;
```

Appendice C

Codice per l'analisi di regressione della liquidità

```
    import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score
import seaborn as sns
import statsmodels.api as sm
from statsmodels.tsa.ar_model import AutoReg
from sklearn.model_selection import train_test_split

# Caricare i dati dal file fornito
file_path = 'tvl_data.xlsx'
data = pd.ExcelFile(file_path)
df = data.parse('Sheet')

# Ricarichiamo i dati mantenendo la prima riga come header e saltando le prime
↪  32 righe di dati
df_filtered = pd.read_excel(
    file_path,
    sheet_name='Sheet',
    header=0,  # Manteniamo la prima riga come header
    skiprows=list(range(1, 32))+[174]  # Saltiamo le righe dalla 2 alla 32 e la
    ↪  174
)

# Selezionare solo le colonne rilevanti e rimuovere i valori mancanti
cleaned_df = df_filtered[['total_liquidity_usd', 'volume',
↪  'deviazione_standard_30gg']].dropna()

# Applicazione della trasformazione logaritmica
cleaned_df['log_liquidity'] = np.log(cleaned_df['total_liquidity_usd'])
```

```
cleaned_df['log_volume'] = np.log(cleaned_df['volume'])
cleaned_df['log_std_dev'] = np.log(cleaned_df['deviazione_standard_30gg'])

# Creazione dei lag per liquidità e volume
cleaned_df['log_liquidity_lag1'] = cleaned_df['log_liquidity'].shift(1)
cleaned_df['log_volume_lag1'] = cleaned_df['log_volume'].shift(1)

# Rimozione dei valori nulli generati dagli shift
cleaned_df.dropna(inplace=True)

# Matrice di correlazione
correlation_matrix = cleaned_df[['log_liquidity', 'log_volume', 'log_std_dev',
↪  'log_liquidity_lag1', 'log_volume_lag1']].corr()
correlation_matrix.columns = ['log(Liquidità)', 'log(Volume)', 'log(Deviazione
↪  Standard)', 'log(Liquidità Lag1)', 'log(Volume Lag1)']
correlation_matrix.index = ['log(Liquidità)', 'log(Volume)', 'log(Deviazione
↪  Standard)', 'log(Liquidità Lag1)', 'log(Volume Lag1)']
print(correlation_matrix)

# Crea una maschera per la metà inferiore della matrice
mask = np.triu(np.ones_like(correlation_matrix, dtype=bool), k=1)
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, mask=mask, annot=True, cmap='coolwarm',
↪  fmt='.2f', linewidths=0.5, vmin=-1, vmax=1, annot_kws={"size": 14})
plt.title('Matrice di correlazione')
plt.tight_layout()
plt.show()

# Definire variabili indipendenti e dipendenti
X = cleaned_df[['log_volume', 'log_std_dev', 'log_liquidity_lag1',
↪  'log_volume_lag1']].values
y = cleaned_df['log_liquidity'].values

# Suddivisione in training e test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
↪  random_state=42)

# Aggiungere una costante per l'intercetta
X_train = sm.add_constant(X_train)
X_test = sm.add_constant(X_test)

# Regressione OLS
ols_model = sm.OLS(y_train, X_train).fit()
y_pred_ols = ols_model.predict(X_test)

# Coefficienti OLS
ols_coefs = ols_model.params
ols_const = ols_model.params[0]  # Intercetta
```

```
# Calcolo di R^2
r2_ols = r2_score(y_test, y_pred_ols)

# Plot del confronto tra i modelli
plt.figure(figsize=(7, 6))

# Grafico OLS
plt.scatter(y_test, y_pred_ols, alpha=0.7, label="Valori previsti (OLS)")
plt.plot(y_test, y_test, color='red', label="Perfetta corrispondenza")
plt.title(f"OLS Regression (R² = {r2_ols:.4f})")
plt.xlabel("Log(Liquidity) effettivo")
plt.ylabel("Log(Liquidity) previsto")
plt.legend()
plt.tight_layout()
plt.show()

# Costruzione della formula OLS
feature_names = ['const', 'log_volume', 'log_std_dev', 'log_liquidity_lag1',
↪  'log_volume_lag1']
ols_formula = " + ".join([f"{coef:.4f}*{var}" for var, coef in
↪  zip(feature_names, ols_coefs) if var != 'const'])
ols_formula = f"{ols_const:.4f} + {ols_formula}"

# Stampa la formula
print("\n=== Formula OLS ===")
print(f"Log(L_t) = {ols_formula}")
```

Appendice D

Matrici di correlazione

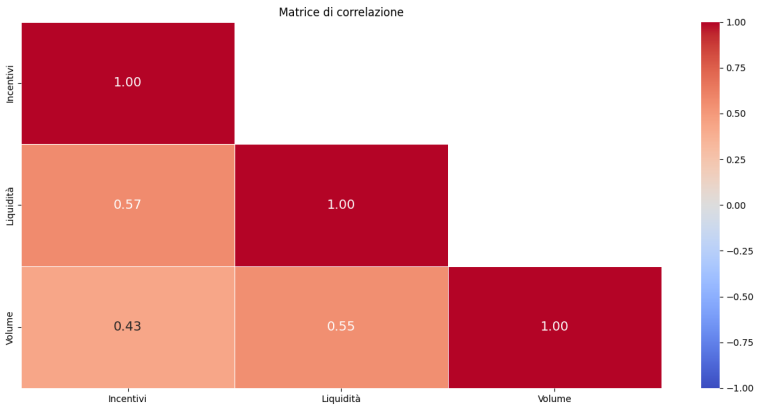


Figura D.1. Matrice di correlazione dati originali

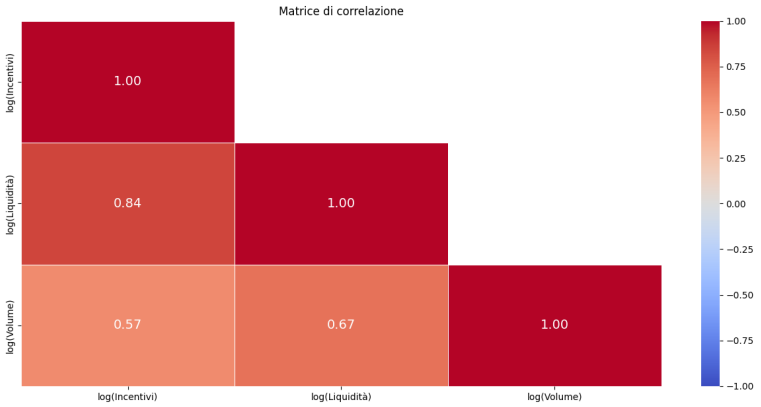


Figura D.2. Matrice di correlazione dati logaritmici

Appendice E

Codice per la risoluzione del modello

```
import numpy as np
import matplotlib.pyplot as plt
from bayes_opt import BayesianOptimization
import scipy.optimize as opt
import pandas as pd
import seaborn as sns

# Impostazione del seed per riproducibilità
np.random.seed(421)

# Parametri iniziali
T = 365
beta = 0.3
rho = 0.9
gamma = 0.95
tau = 0.1
I_tot = 1000000
alpha = 0.01
dec = 0.1
f_opt = 0.01
lambda_penalty = 5  # Penalità più morbida

# Simulazione del volume delle transazioni
mu_volume = 50000
sigma_volume = 50000
volume_t = np.abs(np.random.normal(mu_volume, sigma_volume, T))
mean_transaction_value = 0.001 * np.mean(volume_t)
lambda_min = 10 * mean_transaction_value

# Generazione dei prezzi
mu_price = 2090
```

```
sigma_price = 0.04
price_t = np.random.normal(mu_price, sigma_price, T)

def calculate_deviation(price_t):
    return np.array([np.std(price_t[:t+1]) if t > 0 else 0 for t in range(T)])

deviation_t = calculate_deviation(price_t)

def impermanent_loss(price_ratio):
    return 2 * np.sqrt(price_ratio) / (1 + price_ratio) - 1

price_ratios = price_t / price_t[0]
impermanent_loss_t = np.array([np.mean(impermanent_loss(price_ratios[:t+1])) if
↪  t > 0 else 0 for t in range(T)])

def alpha_t(t, alpha, dec):
    return alpha * np.exp(-dec * t)

def reinforcement_factor(volume_t, liquidity_t):
    return np.tanh(0.01 * (volume_t / np.maximum(liquidity_t, 1e-6)))

def update_volume(T, volume_t, liquidity_t):
    adjusted_volume = np.zeros(T)
    adjusted_volume[0] = volume_t[0]
    for t in range(1, T):
        reinforcement = reinforcement_factor(volume_t[t], liquidity_t[t-1])
        adjusted_volume[t] = volume_t[t] * (1 + reinforcement)
    return adjusted_volume

def utility_function(I_liq_0, incentives, volume_t=None, deviation_t=None,
↪  liquidity_t=None, impermanent_loss_t=None, gamma=None, rho=None):
    # Se chiamata dalla Bayesian Optimization, estrai i valori da kwargs
    if volume_t is None:
        volume_t = globals().get('volume_t', np.zeros(T))  # Usa la variabile
        ↪  globale o default
    if deviation_t is None:
        deviation_t = globals().get('deviation_t', np.zeros(T))  # Default se
        ↪  non passato
    if liquidity_t is None:
        liquidity_t = update_liquidity(T, 0.1, 0.5, -0.3, 0.2, 0.5, 0.5,
        ↪  volume_t, deviation_t, incentives, alpha, dec, I_liq_0)
    if impermanent_loss_t is None:
        impermanent_loss_t = globals().get('impermanent_loss_t', np.zeros(T))
        ↪  # Default se non passato
    if gamma is None:
        gamma = globals().get('gamma', 0.95)  # Usa valore globale o default
    if rho is None:
        rho = globals().get('rho', 0.9)  # Usa valore globale o default
```

```
    # Calcolo della funzione di utilità
    utility = 0
    for t in reversed(range(T)):
        current_utility = (
            (1 - beta) * f_opt * volume_t[t] - incentives[t] +
            rho * (beta * f_opt * volume_t[t] + incentives[t] -
            ↪  impermanent_loss_t[t])
        )
        penalty = lambda_penalty * (lambda_min - liquidity_t[t]) if
        ↪  liquidity_t[t] < lambda_min else 0
        utility = current_utility - penalty + gamma * utility

    return -utility  # Minimizzazione

def update_liquidity(T, beta_0, beta_1, beta_2, beta_3, beta_4, beta_inc,
↪  volume_t, deviation_t, incentives, alpha, dec, I_liq_0, decay_factor=0.99):
    liquidity_t = np.zeros(T)
    liquidity_t[0] = max(I_liq_0, 1e-6)
    for t in range(1, T):
        safe_incentive = max(incentives[t], 1e-6)
        safe_deviation = max(deviation_t[t], 1e-6)
        liquidity_t[t] = (
            np.exp(beta_0) *
            volume_t[t]**beta_1 *
            safe_deviation**beta_2 *
            volume_t[t-1]**beta_3 *
            liquidity_t[t-1]**beta_4 *
            (1 + alpha_t(t, alpha, dec) * safe_incentive**beta_inc)
        ) * decay_factor  # Applica il fattore di decadimento
    return liquidity_t

lambda_arrival_new = 5  # Tasso di arrivo medio di nuovi LP
mu = 0.1  # Tasso medio di uscita degli LP
time_days = np.arange(1, T + 1)
L_t_values = (lambda_arrival_new / mu) * (1 - np.exp(-mu * time_days))
expected_exits = mu * L_t_values

# Modifica dei parametri iniziali e vincoli
def initial_allocation(params):
    I_liq_0, I_inc_0 = params
    if not np.isclose(I_liq_0 + I_inc_0, I_tot, atol=1e-2):  # Maggiore
    ↪  tolleranza sui vincoli
        print(f"Vincoli non rispettati: I_liq_0={I_liq_0}, I_inc_0={I_inc_0}")
        return np.inf
    liquidity_t = update_liquidity(T, 0.1, 0.5, -0.3, 0.2, 0.5, 0.5, volume_t,
    ↪  deviation_t, np.zeros(T), alpha, dec, I_liq_0)
    loss_t = impermanent_loss_t * liquidity_t * expected_exits
    utility = utility_function(I_liq_0, np.zeros(T), volume_t, deviation_t,
    ↪  liquidity_t, loss_t, gamma, rho)
```

```
    if np.isnan(utility) or np.isinf(utility):
        print(f"Valore anomalo di utilità: {utility} per I_liq_0={I_liq_0},
        ↪  I_inc_0={I_inc_0}")

    return -utility

initial_guess = [I_tot * 0.8, I_tot * 0.2]  # Valori iniziali diversi
bounds = [(0, I_tot), (0, I_tot)]

constraints = [
    {'type': 'eq', 'fun': lambda x: x[0] + x[1] - I_tot},
    {'type': 'ineq', 'fun': lambda x: x[0] - lambda_min}
]

result_init = opt.minimize(initial_allocation, initial_guess, bounds=bounds,
↪  constraints=constraints, options={'ftol': 1e-6, 'maxiter': 1000})

# Definizione dei bounds per l'ottimizzazione
pbounds = {f'incentive_{t}': (0, I_tot) for t in range(T//4)}
pbounds.update({'I_liq_0': (0, I_tot)})

# Creazione del modello di ottimizzazione bayesiana
def bayesian_utility_wrapper(I_liq_0, **incentives):
    incentives_array = np.zeros(T)
    for t in range(T//4):
        incentives_array[t] = incentives[f'incentive_{t}']
    return utility_function(I_liq_0, incentives_array)

optimizer = BayesianOptimization(f=bayesian_utility_wrapper, pbounds=pbounds,
↪  random_state=421)

# Esecuzione dell'ottimizzazione
optimizer.maximize(init_points=100, n_iter=100)

# Estrazione dei migliori incentivi e liquidità iniziale
optimized_incentives = np.zeros(T)
I_liq_0_optimized = optimizer.max['params']['I_liq_0']
for t in range(T//4):
    optimized_incentives[t] = optimizer.max['params'][f'incentive_{t}']

# Applica un fattore di decadimento agli incentivi
decay_rate = 0.99
for t in range(1, T):
    optimized_incentives[t] *= decay_rate ** t

# Normalizzazione per garantire che la somma sia esattamente I_tot -
↪  I_liq_0_optimized
total_optimized_incentives = np.sum(optimized_incentives[:T//4])
```

```
optimized_incentives[:T//4] = optimized_incentives[:T//4] * (I_tot -
↪  I_liq_0_optimized) / total_optimized_incentives
print(np.sum(optimized_incentives[:T//4]))

# Calcolo della liquidità ottimizzata con decadimento
optimized_liquidity_decay = update_liquidity(T, 0.1, 0.5, -0.3, 0.2, 0.5, 0.5,
↪  volume_t, deviation_t, optimized_incentives, alpha, dec, I_liq_0_optimized,
↪  decay_factor=0.99)


# Creazione del DataFrame
df = pd.DataFrame({
    "optimized_incentives": optimized_incentives[:T//4],
    "optimized_liquidity_decay": optimized_liquidity_decay[:T//4],
    "volume_t": volume_t[:T//4]
})

# Calcolo della correlazione
print(df.corr())
correlation_matrix = df.corr()
correlation_matrix.columns = ['optimized_incentives',
↪  'optimized_liquidity_decay', 'volume_t']  # Sostituisci con i nomi
↪  desiderati
correlation_matrix.index = ['optimized_incentives',
↪  'optimized_liquidity_decay', 'volume_t']
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='BuPu', fmt='.2f',
↪  linewidths=0.5, vmin=-1, vmax=1, annot_kws={"size": 14})
plt.title('Matrice di correlazione')
plt.show()


# Grafico degli incentivi ottimizzati, volume delle transazioni e liquidità con
↪  decadimento
plt.figure(figsize=(12, 8))

plt.subplot(3, 1, 1)
plt.plot(range(1, T+1), optimized_incentives, marker='o', linestyle='-',
↪  color='limegreen', label='Incentivi Ottimizzati')
plt.title('Distribuzione Ottimale degli Incentivi nel Tempo')
plt.xlabel('Tempo (Periodi)')
plt.ylabel('Incentivi')
plt.legend()
plt.grid(True)

plt.subplot(3, 1, 2)
plt.plot(range(1, T+1), volume_t, marker='s', linestyle='-', color='cyan',
↪  label='Volume delle Transazioni')
plt.title('Andamento del Volume delle Transazioni nel Tempo')
```

```
plt.xlabel('Tempo (Periodi)')
plt.ylabel('Volume')
plt.legend()
plt.grid(True)

plt.subplot(3, 1, 3)
plt.plot(range(1, T+1), optimized_liquidity_decay, marker='^', linestyle='-',
↪  color='magenta', label='Liquidità Ottimizzata con Decadimento')
plt.title('Andamento della Liquidità nel Tempo con Decadimento')
plt.xlabel('Tempo (Periodi)')
plt.ylabel('Liquidità')
plt.legend()
plt.grid(True)

print(f"Liquidità iniziale ottimale: {I_liq_0_optimized:.2f}")
print(f"Incentivi iniziali ottimali: {I_tot - I_liq_0_optimized:.2f}")

plt.tight_layout()
plt.show()
```

Tale codice con le opportune modifiche può essere utilizzato anche per gli incentivi di lockup.

Appendice F

Grafici dei risultati

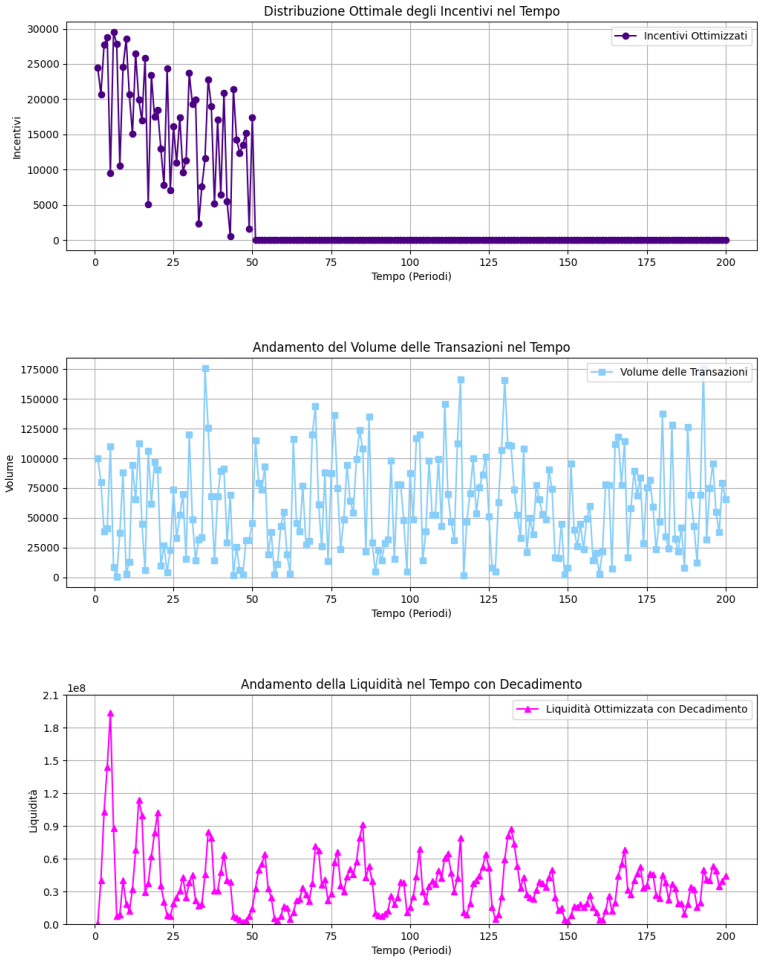


Figura F.1. Risultati con incentivi per early adopter su periodo ristretto

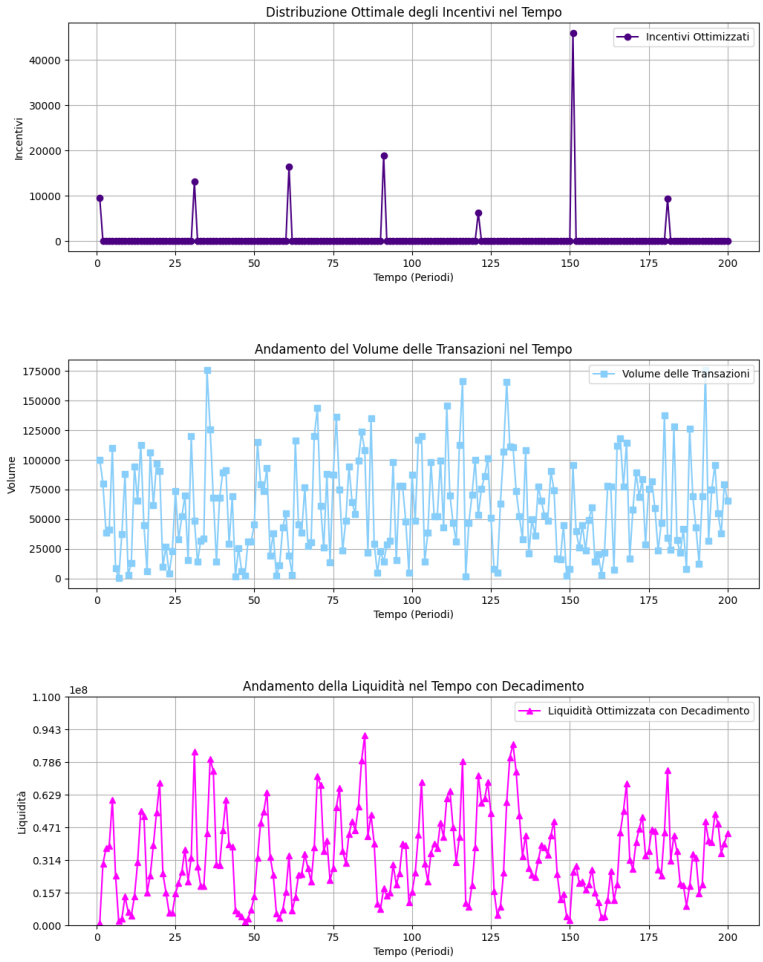


Figura F.2. Risultati con incentivi di lockup su periodo ristretto